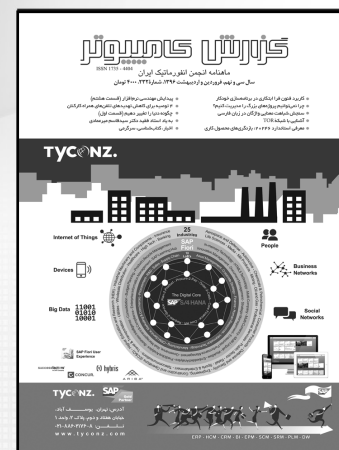


گزارش کامپیوتر

سال سی و نهم، فروردین و اردیبهشت ۱۳۹۶، شماره ۲۳۲، ۴۰۰۰ تومان



صاحب امتیاز: انجمن انفورماتیک ایران

مدیر مسئول و سردبیر: ابراهیم نقیب زاده مشایخ

مقاله ها و گزارش های تالیفی یا ترجمه خود را برای گزارش کامپیوتر بفرستید. مطالب ارسالی را با خط خوانا (یا ماشین شده) بر یک روی کاغذ بنویسید فاصله کافی برای افزودن اصلاحات ویرایشی در بین خط ها و حاشیه در نظر بگیرید. در صورت ارسال مطلب ترجمه شده، یک نسخه از اصل مطلب را نیز ارسال دارید. شکلها باید با روان نویس مشکی ترسیم و به صورت آماده چاپ ارسال شوند. مطالب ارسالی پس فرستاده نمی شوند.

مسئولیت مطالب گزارش کامپیوتر با نویسندگان است و لزوماً نشان دهنده نظر انجمن انفورماتیک ایران نیست. ذکر نام شرکتها و محصولات در مطالب گزارش کامپیوتر برای ارائه اطلاع به خوانندگان است و نباید به معنی تأیید انجمن از آنها تلقی شود.

تمام حقوق به انجمن انفورماتیک ایران متعلق است. نقل مطالب گزارش کامپیوتر با ذکر منبع بلامانع است.

سازمان آگهی ها: مهناز خاوندکار

تلفکس: ۸۸۳۲۸۴۱۷-۲۱

حروفچینی: تارتن سامانه

لیتوگرافی: نگارین پرتو ۷۷۵۳۰۳۰۷

چاپ علوی تهران: ۶۶۷۰۱۵۲۷

خ جمهوری، ۳۰ تیر، کوچه مصری پور، پلاک ۲۵۶

• مقاله

- ۹ کاربرد فنون فرا ابتکاری در برنامه سازی خودکار - علیرضا خلیلیان
- ۱۲ درس هایی در مهندسی نرم افزار: چرا نمی توانیم پروژه های بزرگ را مدیریت کنیم؟
سیدعلی آذرکار
- ۳۷ سنجش شباهت معنایی واژگان در زبان فارسی - نسرين تقی زاده، نفیسه یوری، هشام فیلی
- ۵۳ آشنایی با شبکه TOR - فرزاد آقامحمدی

• گزارش

- ۱۸ ۴ توصیه برای کاهش تهدیدهای تلفن های همراه کارکنان - ابراهیم نقیب زاده مشایخ

• بخش ها

- ۲ اخبار (انجمن، ایران، جهان)
- ۲۰ خواندنی - پیدایش مهندسی نرم افزار (قسمت هشتم) - ابراهیم نقیب زاده مشایخ
- ۴۴ دیدگاه - به یاد استاد فقید دکتر سیدقاسم میرعمادی - سید ابراهیم ابطی
- ۴۶ استاندارد - معرفی استاندارد ۲۰۲۴۶: بازنگری های محصول کاری - سید علی آذرکار
- ۵۸ کتاب شناسی - حقوق بانکداری الکترونیکی با مدیریت فناوری اطلاعات سبز -
سید ابراهیم ابطی
- ۶۳ گوناگون - چگونه دنیا را تغییر دهیم (قسمت اول) - علی حاجی زاده مقدم
- ۷۳ سرگرمی

اعضای هیئت تحریریه و ویراستاران علمی

- مهندس سید ابراهیم ابطی
- دکتر هدیه ساجدی
- دکتر مجید عزیزاده
- دکتر لطف علی بخشی
- دکتر کامبیز بدیع
- دکتر علی رضا باقری
- مهندس سعید امامی
- دکتر محسن صدیقی مشکنانی
- دکتر محمد صنعتی
- دکتر اسلام ناظمی
- دکتر عباس نوذری دالینی
- دکتر محمد گنج تابش

اخبار انجمن

تبریک سال نو

هیئت مدیره انجمن انفورماتیک ایران، فرا رسیدن سال نو را به جامعه انفورماتیک کشور و به ویژه اعضای گرامی انجمن صمیمانه تبریک گفته، سالی سرشار از بهروزی و تندرستی برای آنان آرزو می‌کند.

دریافت مجوز مجله علمی- پژوهشی

انجمن انفورماتیک ایران با همکاری دانشگاه علامه طباطبائی، مجوز مجله علمی - پژوهشی مطالعات مدیریت کسب و کار - هوشمند را در آخرین روزهای سال ۱۳۹۵ از کمیسیون بررسی نشریات علمی کشور، دفتر سیاست‌گذاری و برنامه‌ریزی امور پژوهشی وزارت علوم، تحقیقات و فناوری دریافت کرد.

توضیحات تکمیلی در مورد این مجله در شماره آینده به اطلاع اعضای گرامی انجمن رسانده خواهد شد.

چاپ دوم کتاب بازنگری در کار

به دنبال استقبال گسترده جامعه انفورماتیک کشور از کتاب بازنگری در کار که در مهرماه سال گذشته از سوی انجمن

انفورماتیک ایران منتشر شد و اتمام چاپ اول این کتاب (۱۰۰۰ نسخه) ظرف مدت چهار ماه، چاپ دوم کتاب در اسفندماه ۱۳۹۵ از سوی انجمن منتشر گردید. چاپ دوم کتاب هم مانند چاپ اول با حمایت مالی شرکت پویا انجام گرفت که بدین وسیله از مدیریت محترم آن شرکت سپاسگزاری می‌شود. علاقه‌مندان برای سفارش کتاب می‌توانند با دفتر انجمن (۶۶۴۱۲۸۶۱ و ۶۶۴۱۲۹۷۶) تماس بگیرند.

چاپ چهارمین شماره مجله علوم رایانشی

چهارمین شماره مجله علوم رایانشی، فصل‌نامه علمی- ترویجی انجمن انفورماتیک ایران، در اردیبهشت ماه جاری منتشر شد. در این شماره ۶ مقاله با عناوین زیر به چاپ رسیده است:

۱- روشی جدید جهت شناسایی بدافزارهای فراریخت با تحلیل ایستای ثبات‌ها و کدهای عملیاتی

۲- بهبود معماری خودالتیامی در معماری سرویس‌گرا با استفاده از منطق فازی

۳- اجتناب از موانع روبات سیار با استفاده از الگوریتم میدان پتانسیل مجازی اصلاح شده

۴- ارائه یک مسیر حرکت مناسب برای چاهک‌های متحرک در جهت کاهش مصرف انرژی

۵- طراحی کمپرسورهای ۴ به ۲ و ۵ به ۲ با توان مصرفی پایین

۶- معیارهای ارزیابی شرکت مجری (پیاده‌ساز) راه‌حل برنامه‌ریزی منابع سازمان در صنعت پخش کشور

۷- نسخه الکترونیکی این مجله در نشانی csj.isi.org.ir در دسترس علاقه‌مندان قرار دارد.

راه‌اندازی کانال خبری انجمن در تلگرام

کانال خبری انجمن انفورماتیک ایران در تلگرام راه‌اندازی شد. از این پس علاقه‌مندان می‌توانند علاوه بر وبگاه انجمن، از طریق نشانی https://telegram.me/isi_news نیز از اخبار انجمن آگاهی پیدا کنند.

تفاهم نامه همکاری

یک تفاهم‌نامه همکاری بین انجمن انفورماتیک ایران و انجمن مدیریت اجرایی ایران به امضاء رسید. هدف از امضاء این تفاهم‌نامه، ارتقاء سطح دانش مدیریتی اعضای هر دو انجمن و همکاری‌های مشترک

دارای مدرک بین‌المللی

- برگزاری دوره در محل سازمان متقاضی
 - ارائه مدرک شرکت در دوره از سوی انجمن انفورماتیک ایران
 - ارائه مستندات آموزشی
- علاقه‌مندان می‌توانند با دفتر انجمن انفورماتیک ایران (۶۶۴۱۲۸۶۱ و ۶۶۴۱۲۹۷۶) و یا نشانی پست الکترونیکی member@isi.org.ir تماس بگیرند.

اخبار شاخه‌های دانشجویی

شاخه دانشجویی انجمن انفورماتیک ایران در دانشکده ریاضی، آمار و علوم کامپیوتر دانشگاه تهران، دو سخنرانی علمی به شرح زیر در محل دانشکده برگزار کرد:

۱- آزمون و اشکال‌زدایی نرم‌افزار

زمان: ۲۸ فروردین ۱۳۹۶
سخنران: آقای علیرضا خلیلیان، دانشجوی دکتری مهندسی کامپیوتر در دانشگاه اصفهان و مدرس دانشگاه

۲- افسانه‌های سنو

زمان: ۴ اردیبهشت ۱۳۹۶
سخنران: آقای مهندس رضا شیرازی مفرد، کارشناس ارشد هوش مصنوعی از دانشگاه صنعتی امیرکبیر تهران و مدیرعامل شرکت تندیس تلاش و تفکر

اخبار ایران

چهارمین کنفرانس فناوری اطلاعات، کامپیوتر و مخابرات

چهارمین کنفرانس ملی فناوری اطلاعات، کامپیوتر و مخابرات در ۴ خرداد ۱۳۹۶ در شهر مشهد توسط دانشگاه تربیت مدرس برگزار خواهد گردید. محورهای اصلی این کنفرانس عبارتند از: مخابرات و ارتباطات

فناوری اطلاعات» انجمن انفورماتیک ایران آمادگی برگزاری دوره‌های زیر را برای کارشناسان سازمان‌های مختلف در محل آن سازمان دارد.

- ۱- کارگاه نیم روزه تدوین توافقنامه‌های سطح خدمات
- ۲- کارگاه یک روزه طراحی و استقرار پیشخوان خدمات
- ۳- کارگاه نیم روزه طراحی کاتالوگ خدمات
- ۴- کارگاه یک روزه ابزارهای مدیریت خدمات فناوری اطلاعات
- ۵- کارگاه نیم روزه برون‌سپاری خدمات و محصولات فناوری اطلاعات

- ۶- کارگاه نیم روزه همسوسازی IT با کسب و کار (IT Alignment) از طریق COBIT5
- ۷- کارگاه یک روزه راهبری فناوری اطلاعات (IT Governance) از طریق COBIT5
- ۸- کارگاه یک روزه راهبری فناوری اطلاعات (IT Governance)

- ۹- کارگاه نیم روزه مدیریت تدوam کسب و کار ISO2012:22301
- ۱۰- کارگاه نیم روزه راهبری فناوری اطلاعات و ITBSC

- ۱۱- کارگاه نیم روزه تدوین راهبرد خدمات فناوری اطلاعات
- ۱۲- کارگاه یک روزه مدیریت خدمات فناوری اطلاعات
- ۱۳- کارگاه سه روزه ITIL V3 Foundation 2011

- ۱۴- کارگاه یک روزه مدیریت مخاطرات فناوری اطلاعات
- ۱۵- کارگاه سه روزه دوره جامع COBIT5
- ۱۶- کارگاه نیم روزه توسعه نیازمندی‌ها مبتنی بر چارچوب CCMI-ACQ و استاندارد ISO/IEC29148:2011

- ۱۷- دوره دو روزه اسکرام مستر حرفه‌ای ویژگی‌های این کارگاه‌ها عبارتند از:
 - مشارکت گروهی، بحث و انجام سناریو، حل نمونه آزمون‌های امتحانی
 - استفاده از مثال‌های مرتبط با حوزه فعالیت هر کسب و کار
 - برگزاری دوره حداقل توسط دو مدرس

برای توسعه علم و دانایی در چارچوب موضوع فعالیت‌های هر دو انجمن است.

به موجب این تفاهم‌نامه، تعهدات مشترک زیر برای هر دو انجمن در نظر گرفته شده است:

- ۱- عضویت افتخاری هر انجمن در انجمن دیگر به‌عنوان عضو حقوقی
- ۲- عضویت افتخاری اعضاء هیئت مدیره هر انجمن در انجمن دیگر به‌عنوان عضو حقیقی (در صورت تمایل آنان و در دوره عضویت آنان در هیئت مدیره)
- ۳- معرفی هر انجمن در وبگاه انجمن دیگر و امکان لینک از وبگاه هر انجمن به وبگاه انجمن دیگر

- ۴- مشارکت در انجام طرح‌های علمی و پژوهشی مشترک که با توافق طرفین تعریف می‌گردد

- ۵- برگزاری انواع سمینارها و کارگاه‌های آموزشی مشترک

- ۶- برگزاری جلسات مشترک سالیانه فی‌مابین اعضاء هیئت مدیره هر دو انجمن

- ۷- حضور رایگان اعضاء هر انجمن در سمینارها و کارگاه‌های آموزشی که به صورت رایگان توسط انجمن دیگر برگزار می‌گردد.

- ۸- ارائه ۳۰٪ تخفیف در پرداخت حق عضویت به کلیه اعضاء حقیقی و حقوقی هر انجمن که تمایل دارند در انجمن دیگر عضو گردند. (پس از دریافت معرفی‌نامه از انجمن دیگر)

- ۹- ارائه تا ۳۰٪ تخفیف به اعضاء هر انجمن (با توافق طرفین)، برای ثبت‌نام در کلیه کارگاه‌ها، سمینارها، همایش‌ها و سایر مواردی که توسط انجمن دیگر برگزار می‌گردد.

- ۱۰- اطلاع‌رسانی اخبار مهم هر انجمن به انجمن دیگر از طریق ارسال پیام، ایمیل، و معرفی در وبگاه، نشریات، خبرنامه‌ها و دیگر امکانات اطلاع‌رسانی

برگزاری دوره‌های تخصصی برای سازمان‌ها

گروه تخصصی «راهبری و مدیریت خدمات

- موبایل
- آینده پژوهی در فناوری اطلاعات و ارتباطات
- فناوری اطلاعات
- مهندسی نرم افزار
- هوش مصنوعی
- شبکه های کامپیوتری
- علوم کامپیوتر
- سخت افزار و الکترونیک و رباتیک
- امنیت
- آموزش الکترونیکی
- تجارت الکترونیکی
- شبکه های اجتماعی
- چند رسانه ای
- مهندسی پزشکی
- علاقه مندان برای دریافت اطلاعات بیشتر می توانند به وبگاه کنفرانس به نشانی www.itetconf.ir مراجعه کنند. لازم به ذکر است که انجمن انفورماتیک ایران جزء حامیان علمی و معنوی این کنفرانس است.

اخبار جهان

ماهواره اپل

بنابر گزارش ها، شرکت اپل دست اندرکار ارسال ماهواره ای است که پرتوهای دسترسی به اینترنت پهن باند را به زمین می تاباند. اپل به تازگی دو تن از مدیران بخش ماهواره گوگل را به استخدام خود در آورده و تیم سخت افزاری جدیدی را درون شرکت تشکیل داده است. این دو نفر، جان فین ویک، سرپرست عملیات فضایی گوگل و مایکل ترلا، رئیس بخش مهندسی ماهواره گوگل بودند.

این دو نفر اکنون زیر نظر گرگ دافی، بنیان گذار شرکت «دراپ کم» که پیش از آن ها به اپل پیوسته بود کار می کنند و پیش بینی می شود که آن ها سرگرم کار بر روی پروژه محرمانه ای برای طراحی و ساخت ماهواره ای برای اپل باشند.

اقدام دیگری که نشانگر علاقه فزاینده اپل در این زمینه است، مشارکت با بوئینگ در پرتاب بیش از ۱۰۰۰ ماهواره مدار پائین است. اپل مذاکراتی را با بوئینگ درباره سرمایه گذاری یا مشارکت در این پروژه به عمل آورده است. علاقه مندی اپل به سرمایه گذاری در چنین پروژه هایی قابل درک است. دستیابی سریع تر و با تأخیر کمتر به اینترنت مطمئناً به آینده این شرکت کمک خواهد کرد زیرا باعث افزایش تقاضا برای آیفون و کمک به فناوری بزرگ بعدی، یعنی خانه های متصل به شبکه، خواهد شد.

«اینترنت از آسمان» مدت ها است که رویای شرکت های فناوری بزرگی چون گوگل و فیسبوک بوده است. اما پرتاب تعداد زیادی ماهواره، بسیار پرهزینه و پر مخاطره است و تا کنون باعث کنار رفتن بسیاری از شرکت های ماهواره ای از این حوزه کسب و کار شده است. آی تی نیوز، ۲۱ اپریل ۲۰۱۷

پیش بینی استقبال گسترده سازمان ها از گالکسی اس ۸

پیش بینی می شود که مشتریان سازمانی استقبال گسترده ای از تلفن های گالکسی اس ۸ و اس ۸ پلاس سامسونگ به عمل آورند. این تلفن ها که امروز به بازار عرضه شدند، تمرکز زیادی بر بهره وری کارکنان و مسائل امنیتی دارند.

تحلیل گران آی دی سی پیش بینی می کنند که میلیون ها مشتری سازمانی یکی از این دو دستگاه را خریداری کنند. یکی از دلایل این پیش بینی، خلأ به وجود آمده پس از جمع آوری تلفن های نوت ۷ است. آن تلفن ها هم بر کاربردهای سازمانی تمرکز داشتند.

تحلیل گران پیش بینی می کنند که ۶۵٪ خریداران تلفن های اس ۸، حداقل بخشی از زمان از آن ها برای منظوره های کاری استفاده خواهند کرد. سامسونگ برای فروش ۶۰ میلیون دستگاه تلفن های اس ۸ و اس ۸ پلاس، هدف گذاری کرده است. هدف گذاری سامسونگ برای فروش تلفن های گالکسی

اس ۵۲/۷ میلیون دستگاه بود. قیمت تلفن های اس ۸ از ۷۲۰ دلار شروع می شود و تلفن های اس ۸ پلاس، ۱۰۰ دلار گرانتر هستند. هر دو تلفن تحت سیستم عامل اندروید ۷ کار می کنند.

هر دو تلفن اس ۸ متکی بر سیستم امنیتی و مدیریتی «ناکس» هستند که سامسونگ ظرف سه سال گذشته آن را ترویج کرده و بهبود داده است. بزرگ ترین جاذبه ناکس برای مدیران فناوری اطلاعات این است که ناکس توسط ده ها بُن سازه نرم افزارهای مدیریت سیار سازمانی (EMM) پشتیبانی می شود. ناکس همچنین می تواند برای رمزگذاری پیشرفته برنامه های (app) که بر روی تلفن اجرا می شوند به کار گرفته شود. در حال حاضر، امن ترین تلفن ها در بوم سامانه اندروید، توسط سامسونگ و بلکبری تولید می شوند.

سامسونگ همچنین دستگاه کوچکی را به نام DeX به قیمت ۱۵۰ دلار عرضه کرده که به دارندگان تلفن اس ۸ امکان می دهد که به صفحه نمایش بزرگتری وصل شوند و همانند یک رایانه رومیزی با آن کار کنند. این دستگاه شامل یک صفحه کلید کامل و موش است و به درد کارمندی می خورد که زیاد سفر می کنند و نیاز آنان به خریداری لوحه (تبلت) یا رایانه قابل حمل را برطرف می سازد.

سامسونگ اعلام کرده است که تلفن های اس ۸ فرایندهای آزمایشی تازه ای را برای امنیت باتری انجام داده اند و مانند تلفن های نوت ۷ خطر گرم شدن بیش از حد باتری ندارند.

آی تی ورلد، ۲۱ اپریل ۲۰۱۷

ارتباط صوتی ماشین به ماشین

ارتباط صوتی ماشین ها با یکدیگر در داستان های علمی-تخیلی وجود داشت. اما اکنون در هندوستان در یکی از خطوط اتوبوس رانی به منظور تسهیل سوار شدن مسافران به کار گرفته شده است. مسافران در این خط اتوبوس رانی به جای آن که هنگام

ابتدائی از یک ماده هادی انعطاف پذیر شده اند که برخی عقیده دارند انقلابی در طراحی و ساخت باتری ها، حسگرها و تراشه ها به وجود خواهد آورد.

این پردازنده که تنها با ۱۱۵ ترانزیستور ساخته شده، هنوز خیلی ابتدائی است اما نخستین گام به سوی تولید ریزپردازنده ها بر پایه نیمه هادی های دوبعدی محسوب می شود. مواد دو بعدی از مزیت انعطاف پذیری برخوردارند و این بدان معنی است که می توانند راحت تر در تجهیزات پوشیدنی یا حسگرها به کار برده شوند و احتمال شکسته شدن آن ها را کاهش دهند.

نیمه هادی ها و صفحات نمایش امروزی با وجودی که خیلی نازکند اما هنوز برای کارکردن به ویژگی های سه بعدی موادی که از آن ها ساخته شده اند اتکاء دارند. اگر یک تراشه سیلیکونی را خم کنید می شکند اما مواد دوبعدی نظیر گرافین که در ساخت ریزپردازنده جدید به کار رفته، واقعاً دوبعدی هستند و از بلورهایی به نازکی یک لایه از اتم ها یا مولکول ها ساخته شده اند و بدین جهت، خاصیت انعطاف پذیری دارند.

ریزپردازنده های امروزی اینتل در قالب کلمات ۶۴ بیتی روی داده ها عمل می کنند و می توانند صدها یا حتی هزاران دستورالعمل مختلف را درک کنند. این ریزپردازنده ها شامل صدها میلیون ترانزیستور هستند.

اما ریزپردازنده جدید که توسط پژوهشگران دانشگاه وین ساخته شده تنها می تواند در هر لحظه بر روی یک بیت از داده ها عمل کند و تنها از یک مجموعه چهار دستورالعملی (NOP، LDA، AND، OR) استفاده می کند. مداری که برای ساخت آن به کار رفته ۲ میکرومتر پهنا دارد که این ۱۰۰ برابر بزرگ تر از آن هایی است که در آخرین پردازنده های اینتل و ARM یافت می شود.

پژوهشگران می گویند این تازه آغاز راه است و با کار بیشتر، پیچیدگی ریزپردازنده های جدید بیشتر و اندازه آن ها کوچک تر خواهد شد. ۴۶ سال طول کشید تا اینتل از پردازنده ۴ بیتی ۴۰۰۴ با ۴۶ دستورالعمل به معماری

چند هفته پیش، ابر آمازون برای ۴ ساعت از کار افتاد و این خطر به خوبی خود را نشان داد. هر چند آمازون موفق به راه اندازی دوباره سرویس شد، اما قطع خدمات و وبگاه های مشتریان باعث خسارت های مالی و اعتباری زیادی به آنان شد. چند روز بعد، آمازون اعلام کرد که خطای انسانی عامل آن بوده است. چند کارساز (server) به طور تصادفی از مدار خارج شده بودند و این اشتباه ساده بر روی دیگر کارسازها اثر گذاشته و مشکل بزرگی را به وجود آورده بود.

به عقیده کارشناسان، سازمان ها باید از قراردادن تمام خدمات خود بر روی یک ابر عمومی خودداری کنند و بنابر آن ضرب المثل قدیمی، تمام مرغ ها را در یک سبد نگذارند. یک راه حل مطمئن برای داشتن کسب و کار پیوسته و بدون وقفه، استفاده از ابر خصوصی و یا چند ابر به طور همزمان است که در صورت از کار افتادن یکی، بقیه بتوانند فوراً واکنش نشان دهند.

در حال حاضر، HPE تنها شرکتی است که به سازمان ها کمک می کند تا بنسازهای چند ابری را تعریف و پیاده سازی کنند. در این رویکرد که غالباً هزینه کمتری خواهد داشت، شرکت هایی که قبلاً از ابر عمومی استفاده می کردند به استفاده از ترکیبی از ابر عمومی و ابر خصوصی روی می آورند. HPE به شرکت ها کمک می کند تا ترکیب درستی از فناوری اطلاعات سنتی و ابر خصوصی و عمومی را تعیین کنند. امنیت، کارایی، هزینه و اطمینان پذیری، عواملی هستند که شرکت ها باید به هنگام تعریف و تعیین معماری فناوری اطلاعات خود، به دقت در نظر گیرند. در یک نظرسنجی، ۶۹ درصد پاسخ دهندگان گفته اند که برای مهاجرت به محیط های چند ابری برنامه ریزی کرده اند.

اینفورولد، ۲۱ مارچ ۲۰۱۷

ساخت ریزپردازنده از مواد انعطاف پذیر

پژوهشگران موفق به ساخت یک ریزپردازنده

سوار شدن، کارتی را بر روی صفحه پوش (اسکن) بارکد قرار دهند و یا از ارتباطات نزدیک میدان مغناطیسی (NFC) استفاده کنند، از تلفن هوشمندشان برای ارسال صوت به تلفن راننده استفاده می کنند. برنامه ای (app) که بدین منظور طراحی شده ملودی ملایمی را برای خاتمه تراکنش ارسال می کند. این برنامه به ارتباط بیسیم نیازی ندارد.

مؤسسه گارتنر آینده خوبی را برای ارتباطات ماشین به ماشین به زبان انگلیسی پیش بینی می کند و معتقد است که مکالمه به زبان طبیعی بین دو ماشین، توسط انسان ها قابل درک است و این باعث بهبود امنیت و استحکام سیستم می شود.

به گفته مدیرعامل شرکت انگلیسی شیرپ که این سیستم صوتی را در هندوستان راه اندازی کرده است، این سیستم داده ها را با استفاده از صوت منتقل می کند و این به هر برنامه، محصول یا سرویسی با میکروفن یا بلندگو امکان راه اندازی ارتباط صوتی ماشین به ماشین را می دهد. او این را «بارکد صوتی» نامیده است.

شیرپ می تواند ۵۰ تا ۱۰۰ بیت اطلاعات را در هر ثانیه ارسال کند. مکالمه می تواند دو طرفه باشد و به وجود شبکه نیاز ندارد.

سرویس اتوبوس رانی که از سیستم شیرپ استفاده می کند در شهر دهلی کار می کند و روزانه نزدیک به ۱۵۰۰۰ سفر انجام می دهد. این خط اتوبوس رانی پیش از این از بلیتی که باید با مدرک شناسایی مسافر مطابقت داده می شد استفاده می کرد. این امر باعث کندی خدمت رسانی شده بود.

کامپیوتررولد، ۲۰ اپریل ۲۰۱۷

رویکرد چند ابری

بسیاری از شرکت ها که در صدد یافتن ترکیب درستی از خدمات ابری برای کسب و کارشان هستند، به سوی راهبرد چندگانه فناوری اطلاعات روی آورده اند. در غیر این صورت، کسب و کار روزانه آن ها با خطرات بالقوه جدی روبرو خواهد شد.

باج خواهی از نت فلیکس

یک رخنه گر ادعا کرده است که نسخه کامل سریال بعدی شرکت نت فلیکس را به دست آورده و درخواست مبلغ نامشخصی باج برای جلوگیری از پخش زودهنگام آن در اینترنت کرده است. سریال مزبور، «Orang Is The New Black» نام دارد.

این رخنه گر که تحت نام مستعار Dark Overload عمل می کند، برای اثبات ادعای خود نخستین قسمت سریال را در یک سرویس اشتراک فایل غیرقانونی بارگذاری کرده است.

نت فلیکس اعلام کرده بود که پخش فصل جدید این سریال را از ۹ جون آغاز خواهد کرد. نت فلیکس اعلام کرده است که یک شرکت کوچک توزیع فیلم که با چند استودیوی بزرگ تلویزیونی کار می کند، مورد حمله اینترنتی قرار گرفته است. این شرکت با اعلام «وضعیت فوق العاده»، از افبی آی و سایر مقامات مسئول درخواست رسیدگی کرده است.

کپی سرقت شده سریال می تواند باعث کاهش رشد مشتریان نت فلیکس و قیمت سهام آن گردد. مبلغ مورد درخواست رخنه گر فاش نشده است.

نت فلیکس برآورد کرده بود که به خاطر این سریال، تعداد مشترکانش از ماه اپریل تا جون، ۳/۲ میلیون نفر افزایش یابد. این مقدار بسیار بیشتر از ۱/۸ میلیون مشترک جدیدی است که این شرکت ظرف ۵ سال گذشته به طور میانگین در این فاصله زمانی جذب کرده است.

نت فلیکس یک شرکت سرگرمی آمریکایی است که در سال ۱۹۹۷ در کالیفرنیا تأسیس شده است. این شرکت که ابتدا فقط به پخش فیلم های ویدیویی می پرداخت از سال ۲۰۱۳ به تولید فیلم و سریال نیز روی آورده است. این شرکت هم اکنون خدمات خود را در ۱۹۰ کشور عرضه می کند و بیش از ۹۸ میلیون مشترک در سراسر جهان دارد.

آسوشیتدپرس، ۲۹ اپریل ۲۰۱۷

می شوند و تبلیغات پیش از باز شدن صفحه آغاز و وبگاه ها هستند.

گوگل همچنین مسدود کردن تبلیغات نامناسب موجود در وبگاه ها را در نظر دارد. این کار باعث می شود تا صاحبان وبگاه ها نیز تبلیغاتی را که در مرورگر کروم مسدود می شوند در وبگاه هایشان قرار ندهند.

اینفوورلد، ۲۰ اپریل ۲۰۱۷

رایانه قابل حمل جدید HP

آخرین رایانه قابل حمل هیولت پاکارد به نام زدبوک (Zbook) که به آن ایستگاه کار سیار هم گفته می شود، از نظر ظرفیت حافظه اصلی و فرعی قابل مقایسه با رایانه مک پرو اپل است اما از جهات دیگری برتری های چشمگیری نسبت به آن دارد.

این رایانه ها دارای ۳ درگاه تاندربولت، حافظه DDR4، پردازنده های زیون اینتل و آخرین واحد پردازش گرافیکی (GPU) انویدیا و AMD می باشند. رایانه های مک پرو در مقایسه، دارای ۲ درگاه تاندربولت، حافظه DDR3، پردازنده گرافیکی قدیمی AMD و پردازنده های زیون اینتل با معماری قدیمی تر هستند که در سال ۲۰۱۳ به بازار عرضه شده اند. با وجود این، رایانه های قابل حمل زدبوک که با نمایشگرهای ۱۴، ۱۵/۶ و ۱۷/۳ اینچی عرضه می شوند، نصف قیمت رایانه های مک پرو را دارند. اپل گفته است که گونه ارتقاء یافته مک پرو را در سال ۲۰۱۸ یا ۲۰۱۹ عرضه خواهد کرد. گونه های فعلی مک پرو آنقدر عقب مانده شده اند که بسیاری به لینوکس یا رایانه های شخصی ویندوز روی آورده اند.

رایانه زدبوک ۱۷ که قدرتمندترین ایستگاه کار جدید هیولت پاکارد است، ۱۵۱۹ دلار قیمت دارد. زدبوک ۱۵ که از نظر برخی ویژگی ها کمی ضعیف تر از زدبوک ۱۷ است، ۱۴۱۹ دلار قیمت دارد. زدبوک استودیو که ۱۳۹۹ دلار است، شبیه زدبوک ۱۵ است اما خیلی نازک تر و سبک تر و در حدود ۲ کیلو وزن دارد.

آی تی ورلد، ۲۱ اپریل ۲۰۱۷

x86 امروزی برسد. با تجربیاتی که در این راه اندوخته شده، شاید بهبود نیمه هادی های انعطاف پذیر، زمان کمتری ببرد.

آی تی ورلد، ۲۱ اپریل ۲۰۱۷

سانسور تبلیغات در گوگل

گوگل درآمد فراوانی از تبلیغات برخط دارد. اما در اقدامی که کمی تعجب آور به نظر می رسد، این شرکت می خواهد ویژگی جلوگیری از تبلیغات را در گونه های رومیزی و سیار مرورگر کروم خود بگنجاند. مرورگر کروم هم اکنون بر طبق آمارها محبوب ترین مرورگر جهان است و گوگل انتظار دارد که با گنجانیدن بندآور تبلیغات به عنوان یکی از ویژگی های کروم، کاربران به جای استفاده از محصولات مشابهی که در بازار است، از بندآور پیش فرض خود کروم استفاده کنند. در این صورت، تنها برخی تبلیغات پذیرفتنی خاص نشان داده خواهند شد.

بندآور تبلیغات تهدیدی برای شرکت های تبلیغاتی است. برای مثال، گوگل هر سال مبلغی را به شرکت Eyeo، تولیدکننده آد بلاک پلاس، می پردازد تا تبلیغات جویشر گوگل از پالایه های آن شرکت بگذرند.

با وجود ناهمخوانی ظاهری ورود گوگل به حوزه بندآوری تبلیغات، معیارهای در نظر گرفته شده توسط این شرکت، بر نوع تبلیغاتی که کاربران مرورگر کروم خواهند دید یا نخواهند دید، تأثیر خواهد گذاشت. تا زمانی که کاربران مرورگر این ویژگی را غیرفعال نکنند، تجربه وگردی آنها توسط گوگل تعیین خواهد شد.

بنابر نظرخواهی انجام شده از ۲۵۰۰۰ کاربر، تبلیغات بالاپر (pop-up) و تبلیغاتی که پیش از باز شدن صفحه آغاز و وبگاه ها نشان داده می شود در بین کاربران تلفن های همراه کمترین ارجحیت را دارند اما کاربران رایانه های رومیزی کاملاً مخالف تبلیغات بالاپر، ویدئوهای که خودبه خود نمایش داده

سامسونگ و اپل، سردمدار بازار تلفن‌های همراه هوشمند

سامسونگ و اپل پیشتازی خود در بازار تلفن‌های همراه هوشمند را در آغاز سال ۲۰۱۷ حفظ کرده‌اند و شرکت چینی هواوی با رشد چشمگیری که به دست آورده، جایگاه خود را در مکان سوم تثبیت نموده است.

بنابر گزارش مؤسسه معتبر آی‌دی‌سی، در نخستین سه ماهه سال ۲۰۱۷، فروش تلفن‌های هوشمند با ۴/۳ درصد رشد به ۳۴۷ میلیون دستگاه بالغ گشته است.

رشد فروش تلفن‌های هوشمند در سال ۲۰۱۶ برای نخستین بار کاهش داشت اما نتایج نخستین سه ماهه سال ۲۰۱۷ نشان می‌دهد که این صنعت به دوران شکوفایی خود بازگشته است.

برپایه مطالعه صورت گرفته، سامسونگ با در اختیار داشتن ۲۲/۸ درصد سهم بازار، پیشتاز است و به دنبال آن، اپل با ۱۴/۹ درصد سهم بازار قرار دارد. فروش تلفن‌های هواوی با ۲۱/۷ درصد رشد روبرو بوده و ۹/۸ درصد سهم بازار را از آن خود کرده است. رتبه‌های چهارم و پنجم نیز در اختیار دو شرکت دیگر چینی، آپو و ویوو می‌باشد.

یاهو نیوز، ۲۸ اپریل ۲۰۱۷

رکورد سودآوری برای هاینیکس

شرکت کره‌ای هاینیکس که به تولید تراشه می‌پردازد اعلام کرد که سودش در سه ماهه نخست سال ۲۰۱۷ با بیش از ۳۰۰ درصد رشد روبرو بوده و این به خاطر تقاضای قوی برای تراشه‌های حافظه، مورد استفاده در تلفن‌های همراه، و افزایش قیمت‌هاست. سودخالص این شرکت در سه ماهه نخست سال جاری میلادی، ۱/۸۹ تریلیون وون (۱/۶۶ میلیارد دلار) و سود عملیاتی ۲/۴ تریلیون وون بوده است. هر دوی این مقادیر برای این شرکت رکورد محسوب می‌شوند.

تقاضای بسیار برای تراشه‌های حافظه NAND درختی که در تلفن‌های هوشمند و سایر ابزارک‌های سیار و کمبود مواد اولیه از سوی دیگر، باعث افزایش شدید قیمت‌ها گشته است. قیمت تراشه‌های DRAM که در رایانه‌های شخصی و کارسازها به کار می‌رود و عنصر کلیدی در فروش شرکت هاینیکس است نیز افزایش یافته است.

فروش شرکت هاینیکس در نخستین سه ماهه سال ۲۰۱۷ با ۷۲ درصد افزایش نسبت به دوره مشابه سال گذشته، ۶/۲۸ تریلیون وون بوده که این هم رکوردی برای این شرکت می‌باشد.

شرکت هاینیکس پس از سامسونگ، دومین تولیدکننده تراشه‌های DRAM در جهان است.

خبرگزاری فرانسه، ۲۵ اپریل ۲۰۱۷

ماشین‌های پرنده شخصی

یک شرکت نوپا (استارت‌آپ) در دره سیلیکان به نام کیتی هاوک که ظاهراً مورد حمایت لری پیج یکی از بنیان‌گذاران گوگل قرار دارد، با انتشار ویدیویی از پیش‌نمون (پروتو تایپ) ماشین پرنده خود اعلام کرد که امسال این ماشین‌ها را عرضه خواهد کرد. در وبگاه این شرکت آمده است: «هدف ما تبدیل رویای پرواز شخصی به واقعیت است. ما اعتقاد داریم که وقتی این رویا به واقعیت پیوندد و هر کس قادر به پرواز شخصی باشد، دنیای نامحدودی از فرصت‌ها به رویش گشوده خواهد شد.»

در ویدیوی منتشر شده، یک هواپیمای تک سرنشین از روی یک دریاچه در مکانی نامشخص در بالای آب به پرواز در می‌آید. بلند شدن و فرود این هواپیما مانند بالگرد، عمودی است و با هشت پروانه به جلو می‌رود. وزن آن در حدود ۱۰۰ کیلوگرم است و با سرعت ۴۰ کیلومتر در ساعت و در ارتفاع ۴/۵ متری پرواز می‌کند.

بنابر اعلان شرکت کیتی هاوک، این

ماشین پرنده تماماً برقی است و پرواز آن در مکان‌های کم‌ازدحام آمریکا قانونی است. برای استفاده از آن به گواهی‌نامه خلبانی نیازی نیست و یک آموزش دو ساعته کفایت می‌کند.

رئیس شرکت کیتی هاوک، سباستین ترون استاد علوم کامپیوتر در دانشگاه استنفورد است. او همان کسی است که پدر ماشین‌های خودران گوگل خوانده می‌شود. شرکت کیتی هاوک اعلام کرده است که افراد می‌توانند با پرداخت ۱۰۰ دلار در لیست انتظار و دریافت تخفیف برای خریداری این وسیله نقلیه تازه قرار گیرند. قیمت این ماشین پرنده هنوز اعلام نشده است.

خبرگزاری فرانسه، ۲۵ اپریل ۲۰۱۷

تعداد اعضای لینکدین به نیم میلیارد رسید

شبکه اجتماعی حرفه‌ای لینکدین اعلام کرد که تعداد اعضایش از ۵۰۰ میلیون نفر فراتر رفته است. اعضای لینکدین در ۲۰۰ کشور جهان پراکنده‌اند و به گفتگوهای تخصصی و فرصت‌یابی از طریق این شبکه می‌پردازند.

مایکروسافت سال گذشته لینکدین را به قیمت ۲۶ میلیارد دلار خریداری کرد و این بزرگ‌ترین معامله‌ای است که تاکنون برای یک شرکت شبکه اجتماعی صورت گرفته است.

در زمان خریداری این شرکت توسط مایکروسافت در ماه جون گذشته، لینکدین ۴۳۳ میلیون کاربر ثبت شده داشت.

لینکدین هم اکنون ۱۰ میلیون فهرست کاری فعال دارد، به ۹ میلیون شرکت دسترسی دارد و هر هفته بیش از ۱۰۰ هزار مقاله منتشر می‌کند. به گفته رئیس این شرکت، چنین جامعه تخصصی با این اندازه، تاکنون وجود نداشته است.

خبرگزاری فرانسه، ۲۵ اپریل ۲۰۱۷

ارتباط استفاده زیاد از تلفن همراه با تومور مغزی

برای نخستین بار، دادگاهی در ایتالیا رأی داد که استفاده زیاد از تلفن‌های همراه باعث ابتلای یک فرد به تومور خوش‌خیم مغزی شده است.

روبرتو رومئو، ۵۷ ساله، در دادگاه شهادت داد که مسئولیت‌های کاری او را مجبور می‌کرده که به مدت ۱۵ سال روزی ۳ تا ۴ ساعت از تلفن همراه استفاده کند. او گفت که ابتدا احساس گرفتگی در گوش راستش پیدا کرده و سرانجام در سال ۲۰۱۰، تومور تشخیص داده شده است. خوشبختانه تومور خوش‌خیم بوده اما او شنواییش را از دست داده زیرا جراحان به هنگام عمل مجبور شده‌اند عصب شنواییش را قطع کنند.

پزشکان متخصص، آسیب وارده به فعالیت بدنی رومئو را ۲۳ درصد برآورد کردند و قاضی بر این اساس حکم به پرداخت ماهانه ۵۰۰ یورو از سوی شرکت بیمه به او کرد. مطالعات علمی درباره احتمال خطرات بهداشتی تلفن‌های همراه عمدتاً به این نتیجه رسیده‌اند که در سطحی که اغلب مردم از تلفن‌های همراه استفاده می‌کنند، خطری جدی برای سلامتی انسان وجود ندارد. برخی مطالعات نیز استفاده زیاد از تلفن‌های همراه را خطر آفرین یافته‌اند و بسیاری از خبرگان می‌گویند که هنوز برای ارزیابی دقیق این فناوری نسبتاً جدید، زود است.

یاهو نیوز، ۲۱ اپریل ۲۰۱۷

نسخه جدید نقشه گوگل

گوگل اعلام کرد که نسخه جدیدی از نقشه زمین (گوگل ارت) را آمیخته با داستان‌سرایی و هوش مصنوعی به رایگان عرضه کرده است. مدیر گوگل ارت گفت این هدیه ما به دنیاست.

این نسخه گویا، جنبه آموزشی دارد و اطلاعات با ارزشی را در اختیار کاربران

رایانه، تلفن‌های هوشمند و لوحه‌ها (تبلت) می‌گذارد.

در نسخه جدید گوگل ارت، فرد می‌تواند به اکتشاف کره زمین با راهنمایی صوتی خبرگان بپردازد. هوش مصنوعی گوگل به کاربران اجازه می‌دهد که اطلاعات برخط عمیق‌تری درباره کوه‌ها، کشورها و سرزمین‌هایی که به‌طور مجازی ملاقات می‌کنند کسب کنند. همچنین به کاربران مکان‌های دیگری را که ممکن است بر پایه جستجوهای قبلیش به دیدن آن‌ها علاقه‌مند باشد، پیشنهاد می‌کند.

کاربران نسخه جدید گوگل ارت می‌توانند با استفاده از دکمه 3-D (سه بُعدی) به سراسر جهان پرواز کنند و مناطق شگفت‌انگیزی مانند گراندکنیون را از هر زاویه‌ای که مایل باشند ببینند. به گفته مدیر گوگل ارت، چون بخش اعظم محاسبات در ابر گوگل انجام می‌شود، می‌توان گوگل ارت را بر روی تلفن داشت. بنابراین هر فرد می‌تواند کل دنیا را در جیبش داشته باشد.

نسخه جدید گوگل ارت بر روی مرورگر کروم گوگل و نرم‌افزار اندروید پیاده شده و نسخه مناسب‌سازی شده برای دستگاه‌های اپل و سایر مرورگرها به زودی عرضه خواهد شد. این برای نخستین بار است که گوگل ارت از طریق مرورگر وب قابل دسترسی می‌گردد، نه نصب نرم‌افزار.

آسوشیتدپرس، ۱۹ اپریل ۲۰۱۷

جریمه ۸ میلیون دلاری گوگل

گوگل به دنبال دعوایی دو ساله، سرانجام موافقت کرد ۷/۸ میلیون دلار به شرکت روسی یاندکس بپردازد. این شرکت طبق قانون ضدانحصار، از گوگل به خاطر نصب جویشرگر خود بر روی تلفن‌های اندروید شکایت کرده بود. یاندکس بزرگ‌ترین جویشرگر کشور روسیه است. از این پس دستگاه‌های اندروید که در روسیه مورد استفاده هستند به کاربرانشان این اجازه

را می‌دهند که جویشرگر پیش‌فرض را خودشان انتخاب کنند.

خبرگزاری فرانسه، ۱۸ اپریل ۲۰۱۷

تغییر شکل تولید به کمک چاپگرهای سه بعدی

شرکت اتودسک با همکاری یک کارگاه ریخته‌گری در میشیگان، یک پروژه چاپ سه بعدی را به انجام رساندند که به تولید قاب جدیدی از منیزیوم برای صندلی‌های هواپیما انجامید. این قاب بسیار سبک‌تر از قاب‌هایی است که امروزه در هواپیماها مورد استفاده است و می‌تواند باعث بیش از ۲۰۰ میلیون دلار صرفه‌جویی در هزینه‌های سوخت هواپیما گردد.

اتودسک از نرم‌افزار طراحی سه بعدی خود به نام نت‌فب برای تولید یک مدل هندسی پیچیده برای قاب جدید صندلی مسافران هواپیما که به همان محکمی صندلی‌های قبلی ولی بسیار سبک‌تر است، استفاده کرد. این برنامه طراحی به کمک رایانه (CAD)، فایلی را تولید کرد که مورد استفاده چاپگر سه بعدی برای تولید قاب پلاستیکی صندلی قرار گرفت. این قاب بعداً با مواد سرامیکی پوشش داده شد و در دمای زیاد قرار داده شد تا پلاستیک داخلش تبخیر گردد.

قاب سرامیکی باقیمانده، سپس در کارگاه ریخته‌گری به قاب صندلی منیزیومی تبدیل شد که تنها ۷۶۶ گرم وزن دارد و ۵۶٪ سبک‌تر از صندلی‌های آلومینیومی است که امروزه به کار گرفته می‌شود.

اتودسک ادعا می‌کند که اگر یک شرکت سازنده هواپیما مانند ایرباس، تمام ۶۱۵ صندلی ۱۰۰ جت ۳۸۰٪ خود را با قاب صندلی‌های منیزیومی جایگزین سازد، با احتساب طول عمر متوسط ۲۰ ساله این هواپیماها، ۲۰۶ میلیون دلار در مصرف سوخت، بر پایه قیمت‌های امروزی، صرفه‌جویی خواهد کرد.

کامپیوتر ورلد، ۱۲ می ۲۰۱۷

کاربرد فنون فرا ابتکاری در برنامه سازی خودکار

علیرضا خلیلیان

دانشجوی دکتری نرم افزار، گروه مهندسی نرم افزار، دانشکده مهندسی کامپیوتر، دانشگاه اصفهان

پست الکترونیکی: khalilian@eng.ui.ac.ir

چکیده

سال هاست که دانشمندان تحقیق می کنند چطور برنامه سازی رایانه را به خود رایانه محول کنند! ما به رایانه بگویم چه می خواهیم و او برنامه اش را بسازد و بعد اجرا کند. مقاله حاضر قصد دارد خواننده را فقط با مفاهیم و روش های کلی این مسئله تحقیقاتی آشنا کند و سرنخ هایی از شیوه های حل آن ارائه نماید.

مقدمه

افتادند. یک شاخه از الگوریتم های فرا ابتکاری، الگوریتم های مُلهم از طبیعت^۲ هستند و الگوریتم های هوش توده ای (ازدحامی)^۳ هم زیرمجموعه ای از این الگوریتم ها هستند. الگوریتم های هوش توده ای برای جست و جو در فضای حالت بسیار مناسبند و خواص ویژه ای دارند که آن ها را برای حل مسئله برنامه سازی خودکار^۴ و فائق آمدن بر مشکلات آن مناسب می سازد. این الگوریتم ها هوشمندانه در فضای حالت جست و جو می کنند که جواب های بهتر پیدا شود؛ این الگوریتم ها محدودیت های بسیاری حین جست و جو در نظر می گیرند که سریع تر به جواب برسند. با توجه به مطالب ذکر شده، این مقاله قصد دارد فنیونی را از هوش توده ای که برای حل مسئله برنامه سازی خودکار استفاده شده برشمارد و به طور خلاصه آن ها را مرور نماید. این دسته از فنون را می توان «برنامه سازی توده ای»^۵ نامید.

از ابتدای پیدایش رایانه ها، برنامه سازی توسط انسان انجام شده است و این روند تا سال های آینده هم قطعاً ادامه دارد. اما چندین سال است که محققان به فکر افتادند که چگونه می توان برنامه سازی را خودکار کرد؟ یعنی هدف اینست که آنچه باید انجام شود، با بیانی سطح بالا و انتزاعی به رایانه داده شود و رایانه برنامه ای برای آن مسئله به یک زبان برنامه سازی بنویسد! قطعاً این مسئله سختی است چون دانش و جزئیات حل و پیاده سازی مسئله ها در ذهن انسان هاست، خودکار سازی این کار هوشمندی خاصی نیاز دارد. یک راه برای نیل به این هدف، جست و جو در فضای حالت همه برنامه های ممکن است تا برنامه ای مناسب تعریف مسئله پیدا شود. مشکل این راه حل اینست که فضای حالت همه برنامه های ممکن تقریباً بی نهایت است و جست و جوی کورکورانه در آن غیر عملی است. برای مقابله با این مشکل محققان به فکر استفاده از فنون و الگوریتم های فرا ابتکاری^۱

2- Bio-inspired

3- Swarm intelligence

4- Automatic programming

5- Swarm programming

1- Metaheuristic

الگوریتم‌های ملهم از طبیعت، فراابتکاری و هوش توده‌ای

الگوریتم‌های الهام گرفته از طبیعت بر اساس سامانه‌های زیستی طراحی می‌شوند که خصوصیت‌های این سامانه‌های طبیعی را تقلید می‌نمایند. این سامانه‌ها و نظام‌های طبیعی ساختارهای نسبتاً ساده‌ای دارند، از مؤلفه‌های تصادفی استفاده می‌کنند، پیاده‌سازی رایانه‌ای ساده‌ای دارند و متنوع هستند. این ویژگی‌ها باعث جذابیت آن‌ها به عنوان روشی خوب برای طراحی الگوریتم می‌گردد. اغلب این الگوریتم‌ها برای حل مسائل بهینه‌سازی مورد استفاده قرار می‌گیرند که اصل مسئله NP سخت است و با این الگوریتم‌ها به دنبال به دست آوردن جواب‌های نزدیک به بهینه هستیم. شبکه عصبی مصنوعی^۶، الگوریتم‌های تکاملی^۷، سامانه‌های ایمنی مصنوعی^۸ و هوش توده‌ای از انواع مهم الگوریتم‌های ملهم از طبیعت هستند.

اغلب الگوریتم‌های مذکور و بسیاری الگوریتم‌های دیگر متعلق به خانواده‌ای از الگوریتم‌های هوشمند هستند که آن‌ها را فرا ابتکاری می‌نامند. علت این نام گذاری اینست که این الگوریتم‌ها از چند بخش ابتکاری (مکاشفه‌ای)^۹ تشکیل می‌شوند و کل این اجزا را یک الگوریتم ابتکاری بزرگ تر کنترل می‌نماید و این الگوریتم بزرگ تر فرا ابتکاری نامیده می‌شود. برای نمونه الگوریتم ژنتیک مثال مشهور فرا ابتکاری هاست. در این الگوریتم، اجزای کوچکتری مثل عملیات برش^{۱۰} و جهش^{۱۱} داریم و نیز تابعی موسوم به برازندگی^{۱۲} که طراحی همگی ابتکاری است. در سطح بالاتر، طرز و دفعات و توالی استفاده از عملیات برش و جهش و محاسبه برازندگی را الگوریتم ژنتیک کنترل می‌کند و از این رو فرا ابتکاری نامیده می‌شود.

هوش توده‌ای خانواده‌ایست از الگوریتم‌ها با ویژگی‌های مشترک که همگی ملهم از طبیعت هستند و همگی فرا ابتکاری اند. در این الگوریتم‌ها جمعیتی از عامل‌های ساده وجود دارند که هر عامل رفتار مستقل و تصادفی از خود نشان می‌دهد. هر عامل با محیط و سایر عامل‌ها تعامل می‌کند و اطلاعات منتقل می‌نماید. کل سامانه به صورت نامتمرکز و خودتنظیمی و بدون نظارت و کنترل مرکزی کار می‌کند. عملکرد همه عامل‌ها باعث ظهور رفتار هماهنگ و هوش سراسری می‌گردد. این سامانه‌ها بسیار انعطاف پذیرند و در حل مسایل پیچیده و متنوع کارایی قابل توجهی از خود نشان می‌دهند. به کارگیری جمعیت زیاد باعث مقیاس پذیری، استحکام و وفق پذیری این الگوریتم‌ها می‌شود. بهینه‌سازی کلونی

مورچه^{۱۳}، بهینه‌سازی توده‌ای جزئی (ذره‌ای)^{۱۴}، کلونی زنبورهای مصنوعی^{۱۵}، بهینه‌سازی توده ماهی‌های مصنوعی^{۱۶}، الگوریتم گله‌ای^{۱۷} و کرم شب تاب^{۱۸} از نمونه‌های مشهور الگوریتم‌های هوش توده‌ای هستند.

به گفته میلوناس^{۱۹}، همه توده‌ها هوشمند نیستند. بنابراین او پنج خصوصیت را شناسایی کرد که هر توده باید داشته باشد تا هوشمند شناخته شود:

(الف) همجواری: توده باید قادر باشد محاسبات مکانی و زمانی را انجام دهد.

(ب) کیفیت: جمعیت توده باید بتواند به عوامل کیفی معینی پاسخ دهد؛ مثل میزان غذا، مکان و غیره.

(پ) پاسخ متنوع: منابع باید در بین شاخه‌های مختلف توزیع شوند تا در مقابل تغییرات محیط ایمن‌تر بمانند.

(ت) پایداری و وفق پذیری: اولی یعنی توده نباید با تغییرات محیط رفتار متفاوتی نشان دهد، چون ممکن است قادر به جبران هزینه و انرژی از دست رفته نباشد؛ دومی یعنی اگر تغییر رفتار ارزش هزینه و انرژی مصرفی را داشته باشد، توده باید این کار را انجام دهد. این دو خصوصیت، دو روی یک سکه اند.

فرا ابتکاری‌های کلاسیک برنامه‌سازی خودکار

برنامه‌سازی ژنتیک^{۲۰} و گونه‌های توسعه یافته آن همچون تکامل گرامری^{۲۱} و برنامه‌سازی ژنتیک هدایت شده با گرامر^{۲۲} از روش‌های استفاده شده برای مسئله برنامه‌سازی خودکار هستند که بر مبنای هوش توده‌ای کار نمی‌کنند ولی شباهت‌های بسیاری دارند و بسیار رایجند. برنامه‌سازی ژنتیک الگوریتمی است فرا ابتکاری، تکاملی و ملهم از طبیعت که از الگوریتم ژنتیک برای عمل جست و جوی خود استفاده می‌کند ولی هر یک از اعضای جمعیت آن یک برنامه رایانه‌ای است که با نمایش خاصی مثل درخت اشتقاق^{۲۳} کدگذاری^{۲۴} شده است. در ابتدا تعداد مشخصی از افراد^{۲۵} که هر فرد یک برنامه رایانه‌ایست به تصادف تولید می‌شود. سپس در تکرارهای الگوریتم ژنتیک، عملیات برش و جهش روی افراد اعمال می‌شود تا برنامه‌های جدیدتر تولید شود. مناسب بودن هر برنامه تولید شده با تابع برازندگی اندازه گیری می‌شود و تکرار تا برآورده شدن شرط خاصی ادامه پیدا می‌کند.

13- Ant Colony Optimization (ACO)

14- Particle Swarm Optimization (PSO)

15- Artificial Bee Colony (ABC)

16- Artificial Fish Swarm Optimization (AFSO)

17- Herd

18- Firefly

19- Millonas

20- Genetic Programming (GP)

21- Grammatical Evolution (GE)

22- Grammar-Guided GP (G3P)

23- Parse tree

24- Encode

25- Individual

6- Artificial Neural Network (ANN)

7- Evolutionary Algorithm (EA)

8- Artificial Immune System (AIS)

9- Heuristic

10- Crossover

11- Mutation

12- Fitness

جزئی هندسی^{۳۹} و ترکیب برنامه‌سازی ژنتیک با بهینه‌سازی توده‌ای جزئی^{۴۰} از گونه‌های مورد استفاده برای حل مسئله برنامه‌سازی خودکار بوده‌اند.

رتبه سوم الگوریتم‌ها به فرا ابتکاری‌های برنامه‌سازی توده زنبورها^{۴۱} اختصاص دارد که از الگوریتم کلونی زنبورها^{۴۲} برای جست و جو در فضای حالت برنامه‌های رایانه‌ای در مسئله برنامه‌سازی خودکار استفاده می‌کنند. زنبورها رفتارهایی همچون رقص و ارتباط، ازدواج، تخصیص وظایف، یافتن غذا و تصمیم‌گروهی دارند و این‌ها مبنای طراحی الگوریتم فرا ابتکاری است. کلونی زنبور گرامری^{۴۳} یکی از گونه‌های این خانواده است.

سه فرا ابتکاری برنامه‌سازی توده ماهی‌های مصنوعی، برنامه‌سازی گله‌ای و گله‌دستور زبانی و برنامه‌سازی کرم شب تاب از سایر فراابتکاری‌هایی هستند که از سال ۲۰۱۲ به بعد در حوزه برنامه‌سازی خودکار مورد استفاده قرار گرفته‌اند.

جمع بندی

استفاده از فرا ابتکاری‌های برگرفته از طبیعت روش جست و جوی مناسبی برای غلبه بر مشکلات فضای حالت در بسیاری مسایل کاربردی بخصوص برنامه‌سازی خودکار است. مقاله حاضر صرفاً برخی از مفاهیم لازم را معرفی کرد و با برشمردن الگوریتم‌های فرا ابتکاری مورد استفاده برای برنامه‌سازی خودکار آن‌ها را به جویندگان زمینه‌های تحقیقی معرفی کرد.

مراجع

1. Sipser, Michael. Introduction to the Theory of Computation. 3rd Edition. Boston: Thomson Course Technology, 2013.
2. Dorigo, Marco, Thomas, Stutzle. Ant colony optimization. The MIT Press, Cambridge, Massachusetts, 2004.
3. Kennedy, James F., James Kennedy, Russell C. Eberhart, and Yuhui Shi. Swarm intelligence. Morgan Kaufmann, 2001.
4. Floreano, Dario, and Claudio Mattiussi. Bio-inspired artificial intelligence: theories, methods, and technologies. MIT press, 2008.
5. Yang, Xin-She, Zhihua Cui, Renbin Xiao, Amir Hossein Gandomi, and Mehmet Karamanoglu, eds. Swarm intelligence and bio-inspired computation: theory and applications. Newnes, 2013.
6. Olmo, Juan L., José R. Romero, and Sebastián Ventura. "Swarm-based metaheuristics in automatic programming: a survey." Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 4, no. 6 (2014): 445-469.

39- Geometric PSO (GPSO)

40- Hybrid Genetic Programming with PSO (HGPPSO)

41- Bee Swarm Programming (BSP)

42- Bee Colony Algorithm (BCA)

43- Grammatical Bee Colony (GBC)

برنامه‌سازی ژنتیک ساده سه مشکل عمده دارد: یکی از دست رفتن مقداری اطلاعات حین کدگذاری و عمل جست و جو؛ دوم مسئله تورم کد^{۲۶} و سوم تولید افرادی (برنامه‌هایی) که از نظر گرامری نامعتبرند^{۲۷}. برای حل این مشکل در گونه‌های توسعه یافته برنامه‌سازی ژنتیک از دستور ناپسته به متن^{۲۸} زبان برنامه‌سازی استفاده شده است.

فرا ابتکاری‌های هوش توده‌ای

در بخش‌های قبل الگوریتم‌های فرا ابتکاری که تاکنون در برنامه‌سازی خودکار مورد استفاده قرار گرفته‌اند برشمرده شدند. در این بخش گونه‌ها و جزئیات بیشتری ارائه می‌شود.

بررسی روند مقالات نشان می‌دهد بعد از برنامه‌سازی ژنتیک، رایج‌ترین فرا ابتکاری مورد استفاده برای برنامه‌سازی خودکار، استفاده از بهینه‌سازی کلونی مورچه‌ها بوده است و می‌توان آن‌ها را «برنامه‌سازی مورچه‌ای» نامید. دوریگو برای اولین بار ایده استفاده از رفتار مورچه‌ها را پیشنهاد کرد. بر این اساس رفتار یافتن غذا، ترشح فورمون برای انتقال اطلاعات و مسیریابی مورچه‌ها و سایر رفتار آن‌ها مبنای طراحی خانواده بزرگی از گونه‌های الگوریتم کلونی مورچه‌ها شد.

در حوزه برنامه‌سازی خودکار، برنامه‌سازی کلونی مورچه^{۲۹}، برنامه‌سازی کلونی مورچه مشبک^{۳۰}، برنامه‌سازی مورچه‌ای پویا^{۳۱}، برنامه‌سازی مورچه‌ای تعمیر یافته^{۳۲}، برنامه‌سازی مورچه‌ای تعمیر یافته تقویت شده^{۳۳}، برنامه‌سازی مورچه‌ای کارترین^{۳۴}، برنامه‌سازی مورچه‌ای مبتنی بر دستور زبان^{۳۵} چند هدفه از نمونه‌های مشهور هستند.

هوش توده‌ای جزئی در رتبه دوم بعد از کلونی مورچه قرار دارد که برای طراحی الگوریتم‌های برنامه‌سازی خودکار مورد استفاده قرار گرفته است. حرکت توده پرندگان حین پرواز نمونه طبیعی چنین الگوریتم‌هایی است. معمولاً برای هر فرد (مثلاً پرنده) یک موقعیت مکانی و سرعت حرکت در نظر گرفته می‌شود و این داده‌ها در قالب بردارهایی در فضای هندسی نمایش داده می‌شود و طبق قوانین فرا ابتکاری اصلی هوش توده‌ای جزئی، الگوریتم عمل می‌کند. توده دستور زبانی^{۳۶}، بهینه‌سازی توده درخت‌ها^{۳۸}، بهینه‌سازی توده‌ای

26- Code bloat

27- Syntactically invalid individuals

28- Context Free Grammar (CFG)

29- Ant Colony Programming (ACP)

30- Grid ACP (GACP)

31- Dynamic Ant Programming (DAP)

32- Generalized AP (GAP)

33- Enhanced GAP (EGAP)

34- Cartesian AP (CAP)

35- Grammar-Based AP (GBAP)

36- Multi-Objective GBAP (MOGBAP)

37- Grammatical Swarm (GS)

38- Tree Swarm Optimization (TSO)

درس‌هایی در مهندسی نرم‌افزار

چرا نمی‌توانیم پروژه‌های بزرگ را مدیریت کنیم؟

سیدعلی آذرکار

شرکت مهندسی پدیدپرداز

کمیسیون استاندارد و تدوین مقررات، سازمان نظام صنفی رایانه‌ای استان تهران

پست الکترونیکی: ali.azarkar@pdpsoft.com

مقدمه

مهندسی نرم‌افزار است که گاهی از سوی صاحب‌نظران و متخصصان با عنوان «پدر کیفیت نرم‌افزار» از او یاد می‌شود. همفري در چهارم ژوئيه سال ۱۹۲۷ در ایالت میشیگان به دنیا آمد و در ۸۳ سالگی، در ۲۸ اکتبر ۲۰۱۰ در ایالت فلوریدا درگذشت. او دارای مدرک کارشناسی از دانشگاه شیکاگو و کارشناسی ارشد از انستیتوی فناوری ایلینوی، هر دو در رشته فیزیک بود. علاوه بر آن، او مدرک کارشناسی ارشد در مدیریت حرفه‌ای بازرگانی (MBA) را هم از دانشگاه شیکاگو دریافت کرد.

همفري در سال ۱۹۵۹ به شرکت آی‌بی‌ام پیوست و پس از ۲۷ سال همکاری با این شرکت، در سال ۱۹۸۶ بازنشسته شد. او طی این مدت در سمت‌های «مدیر تیم برنامه‌نویسی» و «معاون فنی توسعه» مشغول به کار بود و ۴۰۰۰ متخصص را در ۱۵ آزمایشگاه این شرکت سرپرستی و مدیریت می‌کرد. تجارب و اندوخته‌های همفري در این شرکت، بعداً شالوده ایجاد مدل‌های مهمی برای توسعه نرم‌افزار شد. همفري بعداً به موسسه مهندسی نرم‌افزار پیوست و فعالیت‌های متعددی را در حوزه کیفیت نرم‌افزار طرح‌ریزی و بنیان نهاد. همکاری او با این موسسه تا اواخر عمر ادامه داشت. او تلاش کرد تجارب سایر حوزه‌های مهندسی را در مهندسی نرم‌افزار به کارگیرد. فعالیت‌های او در معماری مدل

اجرای موفق پروژه‌های نرم‌افزاری، همواره یکی از نگرانی‌های متخصصان و فعالان حوزه مهندسی نرم‌افزار بوده است. در تعریف موفقیت یک پروژه نرم‌افزاری، رعایت ظرف‌های زمانی و هزینه‌ای و میزان تحقق نیازها و اهداف کسب‌وکاری، عوامل تعیین‌کننده‌ای به شمار می‌آیند. مشکلات و موانعی که بر سر راه اجرای موفقیت‌آمیز یک پروژه نرم‌افزاری، به دلیل پیچیدگی‌های خاص که در فرآیند توسعه محصول نرم‌افزاری وجود داشته و دارد، متنوع و گسترده است. این قسمت از نشریه گزارش کامپیوتر که «درس‌هایی در مهندسی نرم‌افزار» نام گرفته، تلاش دارد تا با بازخوانی آموزه‌ها و تجربه‌های ناشی از اجرای پروژه‌های نرم‌افزاری از یک سو، و تحلیل ارکان و فرآیندهای توسعه یک محصول نرم‌افزاری از سوی دیگر، محتوایی را ایجاد کند که به متخصصان داخلی در طراحی و اجرای موفقیت‌آمیز پروژه‌های داخلی کمک کند. با پرهیز از طرح موضوعات نظری، هدف این قسمت آن است مسایل و موضوعاتی را منعکس کند که برای خواننده داخلی آشنا و ملموس باشد.

درباره نویسنده

واتس اس. همفري، از جمله افراد شاخص و تاثیرگذار در حوزه

را روشن نمی‌کرد. سوال بتل از ما (در موسسه مهندسی نرم‌افزار) این بود: «آیا راه بهتری وجود ندارد؟»

۱- مشکل سامانه‌های بزرگ

مشکلاتی که گروه بتل با آن مواجه بودند، مرسوم و فراگیر بود. پروژه سامانه‌های بزرگ همواره با شکست مواجه می‌شود و هرچه آن‌ها بزرگ‌تر باشند، احتمال شکست آن‌ها نیز بیشتر است. به عنوان مثال، سامانه جدید IRS^۶ با پنج سال تاخیر، برای اولین بار در سال ۲۰۰۵ عملیاتی شد، ولی هزینه‌های آن بالغ بر ۲ میلیارد دلار شد. ارزیابی یکی از نهادهای نظامی حاکی از آن است که هزینه توسعه ۷۲ برنامه معمولی موشکی از ۴۰٪ برآورده اولیه فراتر رفته، به طور میانگین ۲۱ ماه تاخیر داشته، و کل هزینه سامانه‌ها نیز به ۲ میلیارد دلار رسیده است.

وضعیت برای سامانه‌هایی که واقعاً بزرگ هستند، از این هم بدتر است. آن گونه که مجله نیویورک تایمز گزارش می‌دهد، دو سوم بزرگ‌ترین سامانه‌های موشکی از ظرف بودجه‌ای خود در سال آخر عبور کرده و در مجموع یک هزینه اضافی بالغ بر ۲۹۶ میلیارد دلار داشته‌اند. این برنامه‌ها، به طور میانگین، با حدود ۲ سال تاخیر انجام شده‌اند.

۲- علت مشکلات

بررسی‌ها نشان می‌دهد که این مشکلات توسعه نوعاً توسط موضوعات فنی حادث نشده، بلکه به میزان زیادی مربوط به مدیریت برنامه است. متأسفانه رایج‌ترین واکنش به مشکلات مربوط به مدیریت برنامه، جایگزینی مدیران برنامه‌ها است. این «فرهنگ مبتنی بر سرزنش»، فرایند اطلاع‌رسانی را مختل کرده و یک سبک مدیریتی دفاعی و غیر شفاف را پرورش می‌دهد. ما سال‌های متمادی است که مدیران را تغییر دادیم، ولی اکنون باید روشن باشد که مشکل مدیران بد نیستند: آن‌ها افراد خوبی هستند که در سمت‌هایی غیرقابل توجیه (از حیث فنی) قرار دارند.

به عنوان مثال، جایگزینی سامانه FBI^۷ اخیراً متوقف شد، در حالی که پروژه سه سال تاخیر داشت و ۱۵۰ میلیون دلار برای آن هزینه شده بود. این برنامه در مجموع ۵ مدیر ارشد فناوری اطلاعات^۸ و ۹ مدیر برنامه داشت. آشکار بود که تغییر مدیران، مشکل این پروژه را حل نکرد. ولی هیچ کدام سامانه‌های خرید را عوض نکردند، پنتاگون را سازمان‌دهی مجدد نکردند یا رویه‌ها را اصلاح نکردند. پروژه‌ها به سمت شکست پیش می‌رفتند. واقعیت این است که امروز پروژه‌های بیشتر و بیشتری نسبت به گذشته با شکست مواجه می‌شوند، و شکست‌ها حتی گران‌تر و دردناک‌تر از قبل هستند.

دیدگاه شایع این است که مدیر برنامه مسئول انجام هر چیزی است که برای انجام کار لازم است. اگر روش مدیریتی یا فن جدیدی نیاز

۶- سامانه نظام مالیاتی ایالات متحده آمریکا
۷- پلیس فدرال ایالات متحده آمریکا

CMM برای نرم‌افزار و توسعه چارچوب‌هایی مانند TSP^۲ و PSP^۳ در زمره کارهای شاخص و برجسته او است. کتاب «مدیریت فرآیند نرم‌افزار» که در واقع جمع‌بندی تجارب او در آی‌بی‌ام و نیز شالوده اصلی توسعه مدل CMM برای نرم‌افزار است، از جمله منابع غنی مهندسی نرم‌افزار به حساب می‌آید.

چارچوب TSP که در این مقاله به آن مکرر اشاره شده، در واقع نتیجه به‌کارگیری فنون جدید مدیریتی برای کارهای دانشی^۴ (از جمله توسعه محصولات نرم‌افزاری و نیز مدیریت پروژه‌های بزرگ نرم‌افزاری) است. خواننده علاقه‌مند می‌تواند جزئیات کامل را در این خصوص از طریق جستجو در اینترنت به‌دست آورد.

چکیده

مدیران در حال تغییر، مقررات تدارکات، رویه‌های خرید، یا تمهیدات قراردادی مشکلات هزینه‌ای و زمان‌بندی مربوط به توسعه سامانه‌های بزرگ را حل نکرده است. این مقاله مشکلاتی را که سازمان‌ها در پروژه‌های بزرگ با آن روبرو هستند، ارائه داده و نشان می‌دهد چگونه یک سازمان دولتی طی یک بازه چندین ساله توانسته توسط TSP پیشرفت کند.

بخش یکپارچه‌سازی سامانه‌های اداره اقیانوس‌شناسی نیروی دریایی ایالات متحده آمریکا^۵، همکاری خود با موسسه مهندسی نرم‌افزار را از سال ۱۹۸۵ شروع کرد. گروه آن‌ها، نرم‌افزاری را برای طیفی از سامانه‌ها که وظیفه آن تامین داده‌های اقیانوس‌شناسی و هواشناسی برای ناوگان نیروی دریایی ایالات متحده تا در سرتاسر دنیا بود، تولید کرده بود. این سامانه‌های عظیم، با چندین ترابایت داده، به شکل ۷×۲۴ فعال بوده و زیرسامانه‌های آن‌ها اطلاعات حساس عملیاتی را برای قریب به اتفاق بخش‌های نیروی دریایی ارسال می‌کردند.

اد بتل (Ed Battle) سرپرست آن بخش (و بعداً، مسئول واحد یکپارچه‌سازی سامانه‌ها) اذعان کرد پیش از آن که آن‌ها کار را با موسسه مهندسی نرم‌افزار شروع کنند، پروژه‌ها اغلب با تاخیر انجام می‌شد، نیازمندی‌ها اغلب به درستی درک نمی‌شد و نادرست بود و علاوه بر آن، هیچ همکاری بین اکثر گروه‌های مستقل وجود نداشت. زمانی که تاریخ تحویل‌های حساس سامانه نزدیک می‌شد، مسئول پروژه کار را از طریق برگزاری جلسات پیشرفت در روزهای دوشنبه، چهارشنبه و جمعه ردیابی می‌کرد. در حالی که این جلسات هم زمان زیادی می‌گرفت و هم فشار زیادی را بر تیم پروژه تحمیل می‌کرد، ولی خیلی وضعیت پروژه

2- Team Software Process

3- Personal Software Process

۴- معادل Knowledge Work است. این اصطلاح برای اولین بار توسط پیتر دراکر (Peter Drucker) که از او با نام «پدر علم مدیریت نوین» یاد می‌شود، ابداع شد. او همچنین از اصطلاح Knowledge Worker نیز برای افرادی که کار دانشی انجام می‌دهند، استفاده کرد. این اصطلاح در ادبیات مدیریتی به «کارگران دانش‌محور»، «دانش‌ورزان»، «کارکنان دانشگر» و «کارکنان دانشی» ترجمه شده است. در این مقاله ترجمه «کارکنان دانشی» به‌کار گرفته شده است.

5- Naval Oceanographic Office (NAVO)

8-Chief Information Officer (CIO)

۱۹۱۱ بیان شد. روش‌های تیلور برای کارگران بی‌سواد و کارهای نسبتاً ساده دستی در گذشته، طراحی شده بود. نوع کار، مهارت‌ها، و روش‌هایی مورد نیاز در اکثر کارهای امروزی، کاملاً با آن زمان متفاوت است؛ ولی روش‌های امروزی مدیریت کماکان از همان اصول کنترل و فرماندهی تیلور تبعیت می‌کنند. متأسفانه، در مورد کارهای نرم‌افزاری و سایر کارهای پیچیده فنی، این روش‌ها برای کنترل هزینه‌ها، زمان‌بندی، و کیفیت پروژه‌ها اثربخشی ندارد. در حالی که مدیران ممکن است برای مدیریت کار شجاعانه تلاش کنند، ولی نمی‌توانند درک کنند کارکنان دانشی چه می‌کنند یا چگونه کار را انجام می‌دهند.

نتیجه غایی این است که مدیران امروزی واقعاً به درستی نمی‌توانند پروژه‌هایی با کار دانشی خود را مدیریت کنند. این بدان معنی است که این پروژه‌ها تحت مدیریت نیست؛ هر کسی می‌داند که پروژه‌های که مدیریت نشوند اغلب شکست می‌خورند. متأسفانه، مدیران عموماً برای شکست‌ها سرزنش می‌شوند، در حالی که مشکل واقعی «نظام مدیریتی» است و نه مدیران. پاسخ، جایگزینی مدیران بالقوه با ظرفیت نیست، بلکه تغییر روش‌های مدیریتی است. به هر حال، مدیران پروژه نوعاً نمی‌دانند چه تغییری باید انجام شود و آگاهانه نسبت به تغییر به یک روش جدید مدیریتی، که عموماً توسط سایر برنامه‌های مشابه استفاده نمی‌شود، تمایلی ندارند.

۵- مدیریت کار دانشی

در باب ملاحظه چگونگی مدیریت کار دانشی، دراکر نتیجه‌گیری می‌کند که: از آن‌جا که مدیران نمی‌توانند به درستی این گونه کارها را مدیریت کنند، کارکنان دانشی باید خودشان را مدیریت کنند. در حالی که مدیران ادغان دارند افراد خودشان را در مدیریت خود درگیر کرده‌اند، ولی واقعیت این است که درگیری از مسئولیت‌پذیری متفاوت است. برای این که کارکنان دانشی به درستی خود را مدیریت کنند، آن‌ها باید در مورد روش‌های مدیریت فردی و تیمی آموزش دیده، و باید برای تولید محصولات خودشان، مذاکره درباره تعهدات خودشان، و برآوردن این تعهدات با محصولات کیفی مسئول باشند. در این حالت، وظیفه مدیر دیگر مدیریت تیم‌های کارکنان دانشی نیست، بلکه راهبری، تشویق، پشتیبانی و سرپرستی آن‌ها است.

کار نرم‌افزاری با این روش کار می‌کند. هنگامی که آن‌ها تلاش می‌کنند اهداف زمان‌بندی مدیریت را برآورده سازند، در آن زمان درباره تعهدات خود با مدیریت مذاکره می‌کنند. تیم‌هایی که احساس می‌کنند شخصاً مسئول کنترل کارهای خود هستند، از وضعیت پروژه آگاهی داشته و داده‌هایی را برای دفاع از تخمین‌های خود دارند. هنگامی مشاهده مشکلات، یا خود آن را

باشد، او باید آن‌ها را در عمل به کار گرفته یا گام‌هایی را که برای این امر لازم است، بردارد. ولی این واقعیت که این‌گونه پروژه‌های بزرگ کماکان با شکست مواجه می‌شوند، بر این دلالت دارد که مدیران برنامه‌ها نمی‌دانند چه باید انجام دهند. به هر حال، ما باید کاری بکنیم و این نکته باید تا کنون روشن باشد که تکیه بر مدیران پروژه برای حل این مشکلات، راه چاره نخواهد بود. این مقاله پیشنهاد می‌کند که چگونه این مشکلات به روشی بیان شوند که مدیران پروژه‌ها بتوانند امروزه آن را پیاده‌سازی کنند.

۳- کار دانشی

به اداره اقیانوس‌شناسی نیروی دریایی توضیح دادیم که مشکلات کار نرم‌افزاری، نشانه‌های اولیه از مشکلاتی بودند که به زودی به تمامی ابعاد کار مهندسی مدرن سرایت خواهند کرد. مدیریت نرم‌افزار از همان آغاز کار دشواری بوده، و دلیل آن ربطی به فناوری ندارد. دلیل آن متفاوت بودن کار نرم‌افزاری است.

برای اغلب کارهای سنتی در گذشته، مدیران در آزمایشگاه یا کارخانه قدم می‌زنند و آن چه در حال انجام بود را مشاهده می‌کردند. این سبک از مدیریت که «مدیریت با قدم زدن»^۹ نامیده شده، یک روش بسیار موثر برای مطلع نگهداشتن مدیریت درباره کار و حضور کارگران در هنگام کار است. به هر حال، مشکل اصلی این سبک مدیریتی آن است که فقط برای کارهایی اثربخش است که شخص می‌تواند از طریق مشاهده کارگرانی که آن را انجام می‌دهند، درک کند. امروزه، پیچیده‌ترین کارهای فنی بیشتر شبیه نرم‌افزار است: حجم زیادی از تلاش خلاقانه در یک رایانه یا در ذهن یک کارگر رخ می‌دهد و تا حد زیادی برای یک ناظر ناآشنا قابل مشاهده نیست.

پیترو دراکر، کسی که برای اولین بار کار دانشی را تعریف کرد، می‌گوید که این کاری است که بیشتر با ذهن انجام می‌شود تا با دست. محصولات، به جای آن که چیزهایی باشند که شما لمس و حس می‌کنید، ایده‌ها هستند. در حالی که این ایده‌ها در نهایت در محصولات فیزیکی متبلور می‌شوند، حجم کار و ارزش واقعی محصول، در تلاش خلاقانه‌ای است که برای توسعه این ایده‌ها و تبدیل آن‌ها به محصولات قابل ارائه در بازار، مورد نیاز است.

۴- مدیریت سنتی

حتی اگر کارگران و حجم زیادی از کارهای آنان به شکل قابل توجهی از ۱۰۰ سال پیش تا کنون فرق کرده باشد، ولی هنوز روش‌های سنتی مدیریت تا حد زیادی مبتنی بر اصول مدیریتی تیلور^{۱۰} است که در کتابش با نام «اصول مدیریت علمی» در سال

9-Management By Walking (MBWA)

۱۰- پدر علم مدیریت (Taylor)

- اصول مدیریتی برای کار دانشی، اساساً با آنهایی که برای مهندسی سنتی به کار می‌رود، تفاوت دارد. پنج اصل مدیریت کار دانشی به شرح زیر است:
- به کارکنان دانشی اعتماد کنید. مدیریت باید به کارکنان و تیم‌های دانشی برای مدیریت خودشان اعتماد کند.
 - تیم‌های قابل اعتماد ایجاد کنید. تیم‌های کار دانشی باید قابل اعتماد باشند. به این معنی که آن‌ها باید بخواهند و بتوانند خودشان را مدیریت کنند.
 - به واقعیت‌ها و داده‌ها اتکا کنید. نظام مدیریتی باید هنگام تصمیم‌گیری به واقعیت‌ها و داده‌ها، به جای جایگاه و ارشدیت، تکیه کند.
 - کیفیت را مدیریت کنید. کیفیت باید بالاترین اولویت سازمان باشد.
 - رهبری کنید. مدیریت باید کارکنان دانشی را رهبری کرده و حمایتی را که برای مدیریت خود لازم دارند، فراهم کند.

پاسخ‌های مبهمی را از قبیل: «کار طراحی را تقریباً کامل کرده‌ام» یا «فقط یکی دو خطا باقی مانده و آزمون را تمام خواهیم کرد» را در جواب ارائه می‌دهند. در حالی که کارکنان دانشی نوعاً اولین کسانی هستند که حس می‌کنند پروژه مشکل زمانی دارد، با این حال، راهی برای توصیف دقیق و روشن وضعیت کار ندارند. آن‌ها به جای این که حرفی بزنند و مخاطره درگیری با انبوهی از انتقادهای مدیریتی را بپذیرند، ترجیح می‌دهند بر روی کار فنی خود متمرکز شوند و مشکل زمان‌بندی را به مدیران خود واگذار کنند.

۷- مشکل شگفتی (غافلگیری)

فرد بروکز^{۱۱} زمانی گفت: «پروژه‌ها زمانی یک روز عقب می‌افتند.» برای نگهداشت پروژه‌ها در چارچوب برنامه زمان‌بندی، همه چیزی که مدیران باید انجام دهند آن است که تیم‌های خود را روزانه از این تاخیرهای «یک روزه» نجات دهند. به هر حال، در کارهای دانشی بزرگ مدیران نمی‌توانند این مشکلات کوچک روزمره را مشاهده کرده و توسعه‌دهندگان هم داده‌ای برای توصیف آن‌ها ندارند. در نتیجه، مدیران نمی‌توانند اقداماتی را برای جبران این تاخیرهای «یک روزه» ندارند. در گذر زمان، این تاخیرهای «یک روزه»، آن قدر زیاد می‌شود که قابل مشاهده است؛ ولی برای این که بتوان کاری برای آن‌ها کرد، خیلی دیر است. به همین دلیل است پروژه‌هایی که حتی توسط مدیران مجرب و کارکشته هم اجرا می‌شود، مشکلات زمان‌بندی، هزینه و کیفیت را با خود دارد. مدیران بازخورد لازم را برای مشاهده مشکل در زمان مناسب و جلوگیری از آن، ندارند. این مانند آن است که گویی در حال راندن یک ماشین با سرعت زیاد در مه غلیظ هستند. زمانی آن‌ها مشکل را می‌بینند که درست در مقابل آن‌ها است و فقط باید با دستپاچگی برای پیشگیری از تصادف، کاری انجام دهند. امروزه، در پروژه‌های سامانه‌های بزرگ، مدیران با سرعت زیاد در یک مه رانندگی می‌کنند، و همواره تصادف اتفاق می‌افتد.

زمانی که مدیران ارشد بیشتری شاهد شکست این پروژه‌ها باشند،

حل کرده یا از مدیریت کمک می‌گیرند. علاوه بر این، زمانی که کارکنان دانشی کارهای خود را اندازه‌گیری و ردیابی کرده و آن‌ها را گزارش می‌دهند، مدیریت داده‌های لازم را به منظور کمک به آن‌ها در رفع مشکلات خواهد داشت. چنین است که کل نظام مدیریتی می‌تواند برای موفقیت برنامه‌های آن‌ها مشارکت کند. هنگامی که تیم‌های کارکنان دانشی از مدیریت، آموزش و پشتیبانی برخوردار باشند، می‌تواند به این روش کار کنند. برای آشنایی با اصول مدیریتی برای کارهای دانشی به جدول ۱ نگاه کنید. سپس آن‌ها به شکل یکنواختی به تعهدات زمانی و هزینه‌ای خود، با ارایه محصولات با کیفیت بالا، عمل می‌کنند. همچنین، این اصول یکسان کار دانشی می‌تواند برای تمامی پروژه‌های مهندسی در یک سازمان اعمال شده، که در نهایت منجر به تولید یک فرآیند مدیریتی قابل اندازه‌گیری و قابل ردیابی برای کار دانشی در کل یک برنامه بزرگ یا حتی کل یک سازمان، شود.

۶- اهداف محیط کار

یکی از بنیادی‌ترین مشکلات در شیوه‌های سنتی مدیریتی آن است که کارکنان و مدیران دیدگاه‌های متفاوتی از موفقیت پروژه دارند. مطالعات نشان می‌دهد که توسعه‌دهندگان محصول، پروژه‌ای را موفق می‌دانند که هم محصول خروجی از حیث فنی جذاب باشد و هم آن‌ها بتوانند با یک تیم یکدست و مشوق کار کنند. این معیارها، مستقل از این که پروژه اهداف زمانی و هزینه‌ای خود را برآورده کرده باشد یا نه، درست است. بر عکس، مدیران پروژه‌هایی را موفق تلقی می‌کنند که اهداف زمانی و هزینه‌ای خود را محقق کرده باشد و خیلی به ماهیت فنی کار یا محیط کاری تیم توجهی ندارند. این تفاوت در اهداف محیط کاری، تاثیر عمیقی بر روی مدیریت برنامه دارد. برای مثال، هنگامی که مدیر برنامه بخواهد بداند یک برنامه بزرگ چه زمانی تمام می‌شود، او این سوال را از مدیران پروژه خواهد پرسید. سپس آن‌ها این موضوع را با اعضای تیم در میان می‌گذارند. با این وجود، اعضای تیم زمان‌بندی را یک مشکل مدیریتی تلقی کرده و

11- Fred Brooks

را توسعه داده و این که یکی از اصول بارز این فرآیند آن است که نظام مدیریتی آن مبتنی بر داده‌های دقیق در سطح عملیاتی است. در TSP، توسعه‌دهندگان داده‌هایی را گردآوری کرده و از آن برای مدیریت کارهای خود بهره می‌برند. همچنین آن‌ها از این داده‌ها برای اندازه‌گیری دقیق وضعیت پروژه در بخش‌های از یک روز استفاده می‌کنند. تیم‌های TSP وضعیت خود را هفتگی مدیریت گزارش داده و مدیریت می‌تواند دقیقاً مشاهده کند که هر بخش از هر پروژه چه وضعیتی دارد. در صورت داشتن اطلاعات دقیق وضعیت، مدیریت می‌تواند مشکلات کوچک هزینه و زمان‌بندی را، قبل از آن که جدی شوند، رویت کند. در این حال، آن‌ها می‌توانند اقدامات بموقع را برای شناسایی و حل مشکلات اتخاذ کنند.

هنگامی که کارکنان دانشی آموزش دیده و بدانند چگونه باید خود را مدیریت کنند، طرح‌های تفصیلی داشته و وضعیت پروژه را به دقت و درستی می‌دانند. همچنین آن‌ها احساس می‌کنند مسئول مدیریت مشکلات خود هستند، و هنگامی که نیازمند کمک باشند می‌توانند از هم تیمی‌های خود، یا اگر ضروری باشد، از مدیریت کمک بگیرند. هیچ فرآیندی مشکلات را حذف نمی‌کند؛ مشکلات یک پیامد طبیعی اجرای کارهای بزرگ و پیچیده است. اما در صورت هشدارهای کافی، اقدامات جبرانی در اکثر موارد همواره امکان‌پذیر بوده، و غالب مشکلات می‌توانند بدون وقوع شکست پیگیری یا حل شوند. هشدار زود هنگام، موضوعی کلیدی است: به همین خاطر است که طرح‌های تفصیلی، سنجه‌های دقیق وضعیت، و مالکیت موضوع در سطح کاری، حیاتی هستند. در کار دانشی، شما فقط زمانی هشدار اولیه را دریافت خواهید کرد که کارکنان دانشی خودشان را مدیریت کنند.

به هر حال، صرف آموزش کارکنان برای چگونگی مدیریت خودشان کافی نیست. بسیاری از مشکلات در کارهای جاری مهندسی توسط خود کارکنان و مدیران ایجاد نمی‌شود، بلکه به این دلیل است که آن‌ها از دانشی که تاکنون به دست آورده‌اند، به درستی استفاده نمی‌کنند. برای استفاده از آن چه آموخته‌اند، آن‌ها باید بدانند چه چیزی را چه موقع و چگونه انجام دهند. برای پروژه‌های بزرگ، یک «فرآیند عملیاتی» اساسی است. مدیریت برنامه، پرداختن به جزئیات بوده و هر گام باید با دقت و به درستی برداشته شود. دقیقاً عین یک خلبان هواپیما، هنگامی که بررسی‌های نهایی قبل از پرواز را انجام داده و سپس یک بازبینی^{۱۲} تفصیلی را دنبال می‌کنند. در عین حالی که آن‌ها با هر گام آشنایی داشته و آن را هزاران بار انجام داده‌اند، ولی مطالعات نشان داده که اکثر حوادث هوایی به این دلیل بوده که دست کم یک مورد از این بازبینی یا فراموش شده یا به درستی اجرا نشده

تاخیرهای زمانی نوعاً در کانون توجه قرار می‌گیرند. علاوه بر این، در یک پروژه بزرگ با چندین همبستگی داخلی، تاخیر در یک قسمت سایر قسمت‌ها را نیز متأثر خواهد کرد. این به معنای آن است که بسیاری از قسمت‌های یک پروژه بزرگ احتمالاً در هم‌زمان با مشکل زمان‌بندی مواجه خواهد شد. در این حالت است که مدیران بخش‌های زیادی از برنامه با یک انتخاب دشوار روبرو خواهند شد: اولین کسی باشم که مشکلات زمان‌بندی را پذیرفته و اعلام کنم یا صبر کنیم تا فرد دیگری با این شکل مواجه شده و آن را اعلام کند؟

۸- مدیریت مبتنی بر سرزنش

متأسفانه، در سامانه‌های جاری، رهبری ارشد تمایل دارد که مدیران را به دلیل مشکلات مدیریتی سرزنش کند. در صورتی که این مدیر اولین فردی باشد که مشکلات را پذیرفته، می‌تواند به راحتی به خاطر مشکلات کل برنامه ملامت و سرزنش می‌شود. جای تعجب نیست که غالب مدیران تصمیم می‌گیرند روی مشکلاتی تمرکز کرده که قادر به حل آن هستند، و منتظر کس دیگری می‌مانند که در سوت خود بدمد. زمانی مشکلات برای رهبری ارشد قابل مشاهده است که برنامه در چنان معضل جدی گرفتار شده که هیچ شانسی برای بازیابی و جبران وجود ندارد. در چنین شرایطی تمام افراد رده بالا غافلگیر می‌شوند.

ترکیب نظام مدیریتی مبتنی بر سرزنش و فقدان سنجه‌های دقیق وضعیت، مشوق ایجاد هم «مدیریت غیرشفاف» و هم «یک بی‌میلی عمومی برای پذیرش مشکلات» است. در برنامه‌های سامانه‌های بزرگ و پیچیده، هر بخش مهم است، مشکلات در هر جا می‌تواند باعث تاخیر هر کسی شود. به همین دلیل است که هر مولفه از کار باید مدیریت و ردیابی شده و باز به همین دلیل است که هر تیم باید تلاش کند تا تمامی تعهدات خود را انجام دهد. باز به همین دلیل است که بدون اطلاعات دقیق وضعیت، تمامی تخمین‌ها و تعهدات در سطح تیم (و به تبع آن در تمامی سطوح بالاتر)، حدس و گمانی بیش نیست. در نهایت این که، به همین دلیل است که چرا با نظام‌های مرسوم مدیریتی امروزی، پروژه‌های بزرگ اغلب تاخیر داشته و از ظرف هزینه‌ای خود خارج می‌شوند.

۹- اداره اقیانوس‌شناسی نیروی دریایی و TSP

هنگامی که ما این نکات را با بتل و همکارانش مرور کردیم، اذعان کرد که همه آن‌ها بسیار منطقی است، ولی تعجب او از این بود که چگونه این موارد به او و دیگر مدیران کمک خواهد کرد تا زمان‌بندی برنامه‌های بزرگ خود را حفظ کنند. توضیح دادیم که موسسه مهندسی نرم‌افزار یک فرآیند کار دانشی که TSP نام دارد

نتیجه‌گیری

شکست‌های پی‌درپی برنامه‌های بزرگ توسعه‌ای نه تنها زمان و هزینه گزافی را خرج می‌کند، بلکه به‌کارگیری فناوری‌های جدید را به تاخیر انداخته و نیروی دفاعی را از دسترسی به ابزارهایی که برای حفاظت از کشور لازم است، محروم می‌سازد. هم‌اکنون باید روشن باشد که صنعت دفاعی ایالات متحده آمریکا فاقد انگیزه لازم برای رفع این مشکل است. به عنوان مثال، یک مدیر میانی اجرایی یک پیمانکار عمده دفاعی اخیراً به من گفت که نمی‌تواند پیشنهاد استفاده از روش‌های توسعه با کیفیت بالا (مانند TSP) را بدهد، چرا که این کار درآمد او را کاهش خواهد داد. به سازمان او زمانی که پروژه طولانی می‌شود، پول پرداخت شده و قراردادهای جدیدی برای رفع مشکلات محصولات ناقص با آن سازمان منعقد می‌شود، اگر این مدیر اجرایی این منبع درآمدی را حذف کند، کارش را از دست خواهد داد. یکی می‌تواند ادعا کند که پاسخ به این وضعیت قراردادهای با مبلغ ثابت خواهد بود؛ ولی این رویکرد نیز چندین بار طی ۵۰ سال گذشته امتحان شده و مشکلی را حل نکرده است.

به طور مشابه، مدیران برنامه نمی‌توانند این مشکل را حل کنند. حتی اگر آن‌ها با TSP آشنا بوده و قانع شده باشند که این کار خواهد کرد، رغبتی به امتحان چیزی قبل از این که توسط دیگر برنامه‌ها به شکل گسترده‌ای استفاده شده باشد، یا این که توسط مدیریت خرید توصیه شده باشد، نخواهد داشت. رویکرد TSP دارای یک تجربه موفق اثبات شده است و می‌تواند این مشکل را همین حالا حل کند. وزارت دفاع ایالات متحده تا آمریکا، یا هر نهاد حاکمیتی، باید TSP یا هر روش محتمل دیگری را به منظور تناسب آن‌ها آزمون کند. سپس آن نهاد باید بهترین روش را برای به‌کارگیری در مدیریت برنامه‌های بزرگ تعیین و توصیه کند که مدیران برنامه‌ها از پیمانکاران خود بخواهند از این روش‌ها استفاده کنند. این نباید یک تلاش زمان‌بر یا پرهزینه باشد. توسعه سامانه‌های بزرگ یک مشکل بسیار حیاتی ملی است که نمی‌توان آن را نادیده گرفت، و این که این صرفه‌جویی‌ها می‌تواند بسیار هنگفت باشد.

قدردانی

از سرکار خانم مرجان طاهری که زحمت تهیه و آماده‌سازی پیش‌نویس این مقاله را متقبل شدند، سپاسگزاری می‌کنم.

منبع

این مقاله از منبع زیر برگرفته شده است:

1- Watts S. Humphrey and James W. Over, Leadership, Teamwork, and Trust: Building a Competitive Software Capability, Addison-Wesley Professional, Reading, MA, 2010

است. تمرکز بر روی کار دقیق، نقش یک فرآیند عملیاتی است: حصول اطمینان از این که هر گام با دقت و به درستی برداشته شده است.

برای بسیاری از کارهای ساده‌ای که همیشه انجام می‌دهیم، به شکلی ناخودآگاه متوجه هستیم که چه چیز را چگونه باید انجام داد. اما برای کارهای جدید و پیچیده و ناآشنا، مانند طرح‌ریزی شخصی، مدیریت دقیق زمان‌بندی، و مدیریت کیفیت بر اساس داده‌های انبوه، گام‌ها روشن نیست. به این معنی که صرف آموزش کارکنان دانشی در باب روش‌های نظری، استفاده از روش‌ها را به شکلی سازگار و صحیح به آن‌ها نخواهد آموخت. برای این منظور، آن‌ها باید یک فرآیند عملیاتی همراه با سنج‌های کیفی و طرح‌های قابل ردیابی، داشته باشند. اما زمانی که کارکنان دانشی به شکل مناسبی آموزش ببینند، بدانند چرا و چگونه خودشان را مدیریت کنند، و یک فرآیند عملیاتی که واقعاً از آن استفاده کنند نیز داشته باشند، آن‌گاه خواهند توانست طرح‌های تفصیلی را ایجاد و دنبال کرده و پیشرفت خود را در مقایسه با این طرح‌ها به دقت ردیابی و گزارش‌دهی کنند.

۱۰- تجربه اداره اقیانوس‌شناسی نیروی دریایی

هنگامی که اداره اقیانوس‌شناسی نیروی دریایی کار را با موسسه مهندسی نرم‌افزار آغاز کرد، آن‌ها از مدل اولیه CMM^{۱۳} استفاده کردند. این مفید بود، ولی پیاده‌سازیش دشوار بود. از سوی دیگر این اداره پی برد که TSP به همراه راهنمایی‌ها که برای مدیریت مناسب پروژه‌های خود نیاز داشتند، مدلی شایسته‌تر است. این اداره همچنین برای یک آموزش سریع آماده شد (آموزش اولیه اعضای تیم یک هفته طول کشید) و تیم‌ها بلافاصله بعد از آن، TSP را آغاز و خودشان را مدیریت کردند.

هنگامی که تیم‌ها شروع به استفاده از TSP کردند، منافع طرح‌ریزی و ردیابی بهتر و کاهش زمان آموزش بلافاصله آشکار شد. بسیاری از سازمان‌ها حتی دریافتند که فقط صرفه‌جویی‌های ناشی از اولین پروژه، آموزش کل تیم و هزینه‌های معرفی آن را جبران می‌کند. سپس تیم می‌تواند به استفاده از آن ادامه داده، بدون آن که سرمایه‌گذاری دیگری برای آموزش لازم باشد.

بعد از به‌کارگیری TSP برای چندین سال، بتل گزارش داد که سطح کیفیت محصولات آن‌ها حدود ۱۰ برابر بهبود یافته و زمان‌های آزمون از ماه‌ها به هفته‌ها کاهش پیدا کرده است. عملکرد زمان‌بندی و هزینه بسیار بیشتر از قبل قابل پیش‌بینی بوده و دیگر به جلسات هفتگی دوشنبه، چهارشنبه و جمعه نیازی نبود. همچنین، مشارکت و هماهنگی تیم به شکل خیلی زیادی بهبود پیدا کرده است. جمع‌بندی نهایی بتل این بود که: «این تنها راه مدیریت پروژه‌های کار دانشی است.»

۴ توصیه برای کاهش تهدیدهای تلفن‌های همراه کارکنان

این روزها مردم سه پنجم وقتشان را، اغلب در محیط کار، بر روی دستگاه‌های همراه می‌گذرانند. اما متأسفانه زیاد به فکر ملاحظات امنیتی نیستند. و این بی‌توجهی باعث به خطر انداختن داده‌های باارزش سازمان یا مشتریان‌شان می‌شود. اگر امنیت دستگاه‌های همراه هنوز برای شرکت شما به صورت یک مشکل در نیامده است، به زودی این اتفاق خواهد افتاد. در زیر، چهار توصیه برای کاهش این تهدیدها ارائه شده است.

است: فقط از شبکه‌های خصوصی مجازی^۵ یا شبکه‌های بیسیم شناخته شده و مطمئن استفاده کنید.

۲- از فروشگاه‌های اینترنتی ناشناخته و نامعتبر برنامه‌های بارگیری نکنید

کارمندان معمولاً مقاومت اندکی در مقابل بارگیری برنامه‌ها از منابع مختلف دارند. اما رفتارهای صحیح و بی‌خطر باید آنقدر تکرار شوند تا به صورت طبیعت ثانویه فرد در آیند. شرکت‌ها باید کارمندان‌شان را تشویق کنند که تنها از فروشگاه‌های اینترنتی معتبر که دارای فرایندهای بازبینی جامع و بررسی‌های امنیتی هستند، نرم‌افزار بارگیری نمایند. شرکت‌ها همچنین باید علاوه بر پایش^۶ تاریخچه و وضعیت برنامه‌هایی که مالک شرکتی دارند، لیستی از برنامه‌های تأیید شده و معتبر را در اختیار کارمندان‌شان قرار دهند.

در غیراینصورت، کاربران در خطر بارگیری برنامه‌هایی قرار می‌گیرند

۱- از شبکه‌های بیسیم (Wi-Fi) نامطمئن دوری کنید

اتصال به شبکه‌های بیسیم نامطمئن در چایخانه‌ها، رستوران‌ها، فرودگاه‌ها و سایر مکان‌های عمومی هیچگاه ایده خوبی نیست. ممکن است راحت باشد، اما کارمندی که با دستگاه‌های همراه‌شان برای انجام کسب و کار به شبکه‌های بیسیم نامطمئن متصل می‌شوند، خطرات زیادی را به جان می‌خرند. نقاط دستیابی^۱ جعلی، خصوصیات شبکه‌های مطمئن و مورد اعتماد را شبیه‌سازی می‌کنند و مدارک رمزگذاری شده گواهی امنیتی^۲ را به دست می‌آورند. این امر به رخنه‌گران امکان می‌دهد تا داده‌ها و گذرواژه‌ها را تغییر دهند یا بربایند. حملاتی که به «مرد میانی»^۳ معروفند، می‌توانند داده‌هایی (از جمله محتویات نامه‌های الکترونیکی و برنامه‌ها^۴) را که از وبگاه‌های نامطمئن و ناامن فرستاده می‌شوند، به چنگ آورند. راه‌حلش ساده

1- access points

2- security certificate

3- man-in-the-middle

4- app

5- VPN

6- monitoring

که می‌دانند به‌طور مرتب باید گذرواژه‌شان را تغییر دهند، معمولاً گذرواژه‌های قوی انتخاب نمی‌کنند و به احتمال زیاد، برای آن که خودشان یادشان نرود، گذرواژه‌ها را روی کاغذ می‌نویسند.

۴- رمزگذاری^{۱۰} و احراز هویت دوعاملی^{۱۱}

سازمان‌ها باید تمام دستگاه‌های همراه را به یک سیستم رمزگذاری قوی مجهز کنند تا در صورت گم شدن یک تلفن یا رایانه قابل حمل، یابنده نتواند قادر به خواندن اطلاعات حساس و محرمانه سازمانی که در آن دستگاه‌ها ذخیره شده است گردد. شرکت‌ها می‌توانند فراتر از شناسه کاربری و گذرواژه، برای هر دستگاهی که به داده‌های سازمانی دسترسی دارد، درخواست احراز هویت دوعاملی کنند. این امر همچنین برخی جنبه‌های مربوط به حفاظت گذرواژه، به ویژه گذرواژه‌هایی که به راحتی حدس زده می‌شوند و استفاده مکرر از گذرواژه‌های مشابه، را حل خواهد کرد.

ترجمه: ابراهیم نقیب‌زاده مشایخ

منبع: اینفوورلد، ۲۰ اپریل ۲۰۱۷

10- encryption

11- two-factor authentication

که در واقع همانند بمب‌های دیجیتالی هستند. این برنامه‌ها در ظاهری قابل اعتماد عرضه می‌شوند اما همراه بدافزارهایی^۷ بارگیری می‌شوند که برای مجرمان رایانه‌ای^۸ امکان دستیابی به داده‌های سازمانی را فراهم می‌سازند. به عنوان یک نمونه، در پائیز سال ۲۰۱۶، بیش از یک میلیون حساب گوگل توسط بدافزاری که به رخنه‌گران اجازه می‌داد تا حتی بدون وارد کردن گذرواژه به داده‌های صاحب دستگاه همراه دستیابی پیدا کنند، لو رفت.

۳- شماره شناسایی شخصی^۹ و گذرواژه قوی

این مسئولیت کارمندان است که از انتخاب گذرواژه‌ها و شماره شناسایی شخصی آسان و راحت خودداری کنند. هر چند رخنه‌گران به تلاش بیشتری برای شکستن گذرواژه‌های پیچیده و طولانی نیاز دارند، اما اغلب مردم هنوز از یک گذرواژه در همه جا استفاده می‌کنند. در همین حال، سازمان‌ها نیز باید در مجبور کردن کارمندان به تغییر هر از چندگاه گذرواژه‌های دستگاه‌هایشان تجدید نظر کنند. به عقیده پژوهشگران، این کار نقض غرض است. زیرا کاربرانی

7- malware

8- cybercriminals

9- PIN

منتشر شد!

پایان کار غول‌ها

نوشته: نیکومیل

ترجمه: ابراهیم نقیب‌زاده مشایخ

قیمت: ۱۰/۰۰۰ تومان

برای تهیه کتاب با دفتر انجمن انفورماتیک ایران

تماس بگیرید ۶۶۴۱۲۸۶۱

فروش اینترنتی در فروشگاه چاره

www.chare.ir



پیدایش مهندسی نرم افزار باربارا لیسکوف و انتزاع داده ها

(بخش هشتم)

نوشته ادگار دی لایت

ترجمه: ابراهیم نقیب زاده مشایخ

پست الکترونیکی: mashayekh@isi.org.ir

پیش گفتار مترجم

در مجله CACM مربوط به ژانویه ۲۰۱۵، مقاله ای توجهم را جلب کرد با عنوان «اشک های دونالد کنوت». این مقاله که توسط توماس هی نوشته شده بود، در واقع انتقادی بود به مطالبی که کنوت در سخنرانی خود در دانشگاه استنفورد بیان کرده بود. در آن سخنرانی، کنوت به شدت به مقاله مارتین کمپبل - کلی با عنوان «تاریخ تاریخ نرم افزار» تاخته بود و از این که او در مقاله اش سطح تاریخ رایانش و علوم کامپیوتر را این قدر پایین آورده به شدت انتقاد کرده بود. کنوت در پایان سخنرانی اش گفته بود که آنقدر از خواندن آن مقاله ناراحت و غمگین شده بود که شیشه های عینکش از اشک های چشمش لک شده بود و او به سختی توانسته خواندن مقاله را به پایان برد.

توماس هی در مقاله اش ضمن تمجید از کنوت و دستاوردهای بی ماندش در حوزه رایانش، از این که کتاب های کمی درباره تاریخ علوم کامپیوتر نوشته شده با او همداستانی کرده اما این کوتاهی را بیشتر متوجه خود دانشمندان این رشته دانسته است. او گفته: «امروز شما می توانید در دانشکده های ریاضی و پزشکی، حقوق و چند رشته دیگر مدرک دکترای بگیرید و موضوع رساله تان، تاریخچه آن رشته باشد. اما چنین امکانی در دانشکده های علوم کامپیوتر وجود ندارد. من دانشکده علوم کامپیوتری را در آمریکا نمی شناسم که کسی را که موضوع رساله دکتریش، جنبه های تاریخی علوم کامپیوتر بوده، استخدام کرده باشد. همچنین کسی را در آمریکا نمی شناسم که به خاطر کارهایش در زمینه تاریخ علوم کامپیوتر در یک دانشکده علوم کامپیوتر تشویق شده یا ارتقاء یافته باشد.»

مقاله ۵ صفحه ای توماس هی بسیار خواندنی است و علاقه مندان به این موضوع را به خواندن متن کامل آن توصیه می کنم. او دارای درجه کارشناسی علوم کامپیوتر از دانشگاه منچستر (۱۹۹۵) و دکترای تاریخ و جامعه شناسی علم از دانشگاه پنسیلوانیا (۲۰۰۳) است. اما غرض از آوردن این مقدمه این بود که در مقاله مزبور به چند کار ارزشمند نیز که در این زمینه انجام شده اشاره گشته که از آن جمله، کتاب «پیدایش مهندسی نرم افزار: از تورینگ تا دایکسترا» نوشته ادگار دی لایت است. عنوان این کتاب برای من که نزدیک ۳۰ سال است درس مهندسی نرم افزار را در دانشگاه تدریس می کنم، جالب بود. جستجویی در اینترنت کردم و دیدم که افراد سرشناسی چون گریدی بوچ و جان رینولدز نقدهای بسیار خوبی درباره این کتاب نوشته اند. نویسنده کتاب، ادگار دی لایت نیز خود دارای درجه کارشناسی ارشد منطق از دانشگاه آمستردام (۲۰۰۹) و دکترای مهندسی نرم افزار از دانشگاه لوون بلژیک (۲۰۰۶) و در حال حاضر پژوهشگر تاریخ رایانش در دوره پسادکتری در دانشگاه فناوری آیندهون هلند است. کتاب را توسط دوستی در کانادا از کتاب فروشی آمازون خریداری کردم و ظرف یکی دو هفته توسط مسافری به دستم رسید. خودم از خواندن آن لذت بردم و آن را به دو تن از همکارانم در دانشکده علوم کامپیوتر دانشگاه تهران نیز نشان دادم و آن ها هم مطالب مطرح شده در کتاب را برای پژوهشگران و متخصصان علوم کامپیوتر و به ویژه دانشجویان این رشته بسیار مفید دانستند. از این رو به ترجمه کتاب اقدام کردم که به صورت یک سلسله مقاله دنباله دار در این نشریه به نظر تان خواهد رسید.

سال‌های نخستین

دی‌لایت: شما در ۷ نوامبر ۱۹۳۹ در لس‌آنجلس به دنیا آمدید و نخستین فرزند از چهار فرزند خانواده‌ای شدید که به قول خودتان «برای تحصیلات ارزش زیادی قائل بودند.» [۱۱۴] می‌شود بیشتر توضیح دهید؟

لیسکوف: پدرم وکیل بود و مادرم با آن که خانه‌دار بود، تحصیلات دانشگاهی داشت. مسیر موفقیت از تحصیلات می‌گذشت، بنابراین انتظار این بود که من هم به دانشگاه بروم. همهٔ افراد خانوادهٔ ما برای رفتن به دانشگاه برنامه‌ریزی کرده بودند. برادر من سال از من کوچک‌تر بود. بنابراین شاید من تشویق بیشتری شده باشم زیرا این یک الگوی متداول در خانواده‌هایی که فرزند اولشان دختر است می‌باشد. من همیشه به خاطر عملکرد تحصیلی خیلی خوب در مدرسه تشویق می‌شدم. و این تشویق، خاص دروس ریاضی و علوم نبود بلکه برای عملکرد تحصیلی‌ام به‌طور کلی بود.

دی‌لایت: شما پنج ساله بودید که جنگ جهانی دوم تمام شد. آیا چیزی به خاطر می‌آورید؟

لیسکوف: چیز زیادی نه، فقط آتش‌بازی جشن پیروزی را به یاد می‌آورم.

دی‌لایت: دههٔ ۱۹۵۰ را چگونه توصیف می‌کنید؟ آمریکا در جنگ پیروز شده بود. بنابراین می‌توان تصوّر کرد که در این نقطه از جهان باید دوران واقعاً خوبی بوده باشد.

لیسکوف: خوب، البته من بچه بودم و چیزی برای مقایسه در اختیار نداشتم. اما به نظرم می‌آید که دوران پر رونقی بود. دنیا در مورد آمریکا خوب فکر می‌کرد و آمریکا به نوعی از نظر فرهنگی پیش‌تاز بود. هر اتفاقی که در آمریکا می‌افتاد، در اروپا نیز گسترش می‌یافت. از نقطه نظر این که با زنان چه رفتاری می‌شد، دوران خوبی نبود. با بازگشت مردان از جنگ، زنان از کارخانه‌ها به خانه فرستاده می‌شدند و جایشان را به مردان می‌دادند. من فکر می‌کنم دههٔ ۱۹۵۰، زمان هم‌نوایی و توافق بود. خانواده‌های آمرانی که زنان در خانه بودند و مردان کت و شلوار پوشیده در محل کار. و هر سال یک ماشین جدید. زندگی خیلی مادی بود. خانوادهٔ ما کمی غیرمعمول بود زیرا ما این کارها را نمی‌کردیم. پدرم خیلی آگاه و مطلع بود. ما دربارهٔ نابخردانه بودن این نوع زندگی، با هم مرتباً صحبت می‌کردیم.

دی‌لایت: پدر و مادر شما آمریکایی بودند؟

لیسکوف: والدین پدرم از اروپا به آمریکا مهاجرت کرده بودند. او یک نسل اولی بود. والدین مادرم در آمریکا به دنیا آمده بودند. مادر مادرم در سانفرانسیسکو به دنیا آمده بود، بنابراین از طرف مادر، ما نسل چندم سانفرانسیسکویی‌ها بودیم. اما خانوادهٔ پدرم از لهستان یا روسیه آمده بودند و در پورتلند در ایالت مین که در ساحل شرقی است ساکن شده بودند.

برای کمک به مستندسازی هر چه بهتر تاریخچهٔ مهندسی نرم‌افزار، گفتگویی را با باربارا لیسکوف^۱، برندهٔ جایزهٔ تورینگ در سال ۲۰۰۸، ترتیب دادم. موضوعات مورد علاقهٔ من در این گفتگو، کارهای پیش‌تازانهٔ او در طراحی سیستم عامل لایه‌ای^۲ و ایده‌های او دربارهٔ انتزاع^۳ بود که عمدتاً در دههٔ ۱۹۷۰ گسترش یافته‌اند. از یک سو، کار تورینگ و منطق ریاضی تنها به صورت اجمالی مورد بحث قرار گرفت زیرا لیسکوف برای پیشبرد پژوهش‌هایش، اتکاء چندانی به آن‌ها نداشته است. اما از سوی دیگر، کارهای دایکسترا و ویرث در زمینهٔ برنامه‌نویسی ساخت‌یافته به‌طور عمیق و با جزئیات بیشتر مورد بحث واقع شد. با وجود این، هر دو مورد درخور توجه هستند. مستندسازی مورد اول، هر چند خلاصه، به اهمیت مستندسازی مورد دوم است زیرا به درک میزان تأثیر و نفوذ تورینگ بر مهندسی نرم‌افزار کمک می‌کند.

ملاقات با باربارا لیسکوف

گفتگوی من با لیسکوف در تاریخ ۱۵ دسامبر ۲۰۱۰ در مرکز فناوری ماساچوست (MIT) صورت گرفت. صدای ضبط شدهٔ این گفتگو، توسط من بر روی کاغذ منتقل و سپس توسط لیسکوف و من ویرایش گردید.

زندگی‌نامهٔ مختصر باربارا لیسکوف

باربارا لیسکوف در ۷ نوامبر ۱۹۳۹ در لس‌آنجلس به دنیا آمده و در سانفرانسیسکو بزرگ شده است. او در سال ۱۹۶۱ مدرک کارشناسی ریاضی را از دانشگاه کالیفرنیا در برکلی دریافت کرد و پس از آن به‌عنوان برنامه‌نویس رایانه ابتدا در شرکت میترا^۴ در بوستن و سپس در دانشگاه هاروارد به کار مشغول شد. پس از به دست آوردن این تجربهٔ کاری، از سال ۱۹۶۳ تا ۱۹۶۸ در دانشگاه استنفورد دانشجوی دکتری جان مک‌کارتی شد و بر روی پایان‌بازی‌های شطرنج پژوهش کرد. او سپس به شرکت میترا بازگشت و به کار در زمینهٔ برنامه‌نویسی سیستم^۵ پرداخت. چهار سال بعد، در سال ۱۹۷۲، او به‌عنوان عضو هیئت علمی به ام‌آی‌تی پیوست و تاکنون در همانجا بوده است. لیسکوف تاکنون به خاطر پژوهش‌هایش چندین جایزه دریافت داشته که از آن جمله می‌توان به جایزهٔ دستاورد انجمن زنان مهندس (۱۹۹۶)، مدال جان فون نویمان انجمن مهندسان برق و الکترونیک (IEEE) در سال ۲۰۰۴، جایزهٔ دستاورد زبان برنامه‌نویسی انجمن ماشین‌های رایانشی (ACM) در سال ۲۰۰۸ و جایزهٔ تورینگ در سال ۲۰۰۸ اشاره کرد.

1-Barbara Liskov
2-layered operating system
3- abstraction
4- Mitre Corporation
5- system programming

دی لایت: آیا سرگرمی خاصی داشتید یا آن که بیشتر به کارهای تحصیلی خود می پرداختید؟

لیسکوف: من سرگرمی های زیادی در دوران جوانی خود داشتم و در دانشگاه به رقص های محلی می پرداختم. دوستانی داشتم که کوهنورد بودند و من معمولاً همراه آن ها می رفتم، هر چند این کار را زیاد دوست نداشتم چون از ارتفاع می ترسیدم. همچنین دوستانی هم داشتم که با آن ها رفت و آمد می کردم. بنابراین، این طور نبود که فقط به درس خواندن مشغول باشم. من به عنوان یک زن جوان، زندگی کاملاً معمولی داشتم، بجز این که در رشته ریاضی درس می خواندم و بعداً هم وارد دوره دکتری شدم که خیلی در آن زمان برای دختران معمول نبود.

دی لایت: شما دوره کارشناسی ریاضی را در سال ۱۹۶۱ تمام کردید. و سپس برای دوره کارشناسی ارشد ریاضی از دانشگاه های برکلی و پرینستون درخواست پذیرش کردید. درست است؟

لیسکوف: بله، دقیقاً یادم نمی آید که از چه دانشگاه های دیگری هم درخواست پذیرش کرده بودم. به ادامه تحصیل فکر می کردم و به همین خاطر از چند جا درخواست پذیرش کرده بودم اما بعد تصمیم گرفتم بهتر است بلافاصله ادامه تحصیل ندهم. به همین خاطر مشغول کار شدم.

دی لایت: در بوستن؟

لیسکوف: بله، زیرا خانواده پدرم آنجا بودند و فکر کردم جالب است که زندگی در آنجا را تجربه کنم.

دی لایت: برای یک اروپایی چه توضیحی می توانید درباره تفاوت بین شرق و غرب آمریکا بدهید؟

لیسکوف: خوب، تفاوت خیلی بیشتر از حالا بود. در آن موقع هم هواپیما بود ولی سفر هوایی خیلی به نسبت درآمدها گرانتر بود. الان همه با هواپیما مسافرت می کنند. به هر حال، ۴۵۰۰ کیلومتر فاصله است. نحوه ای که دره سلیکان توسعه یافته است مطمئناً پدیده بسیار جالبی است. این در آن دوره ای که ما صحبتش را می کنیم واضح نبود زیرا هنوز آغاز نشده بود. البته لس آنجلس همیشه متمایز از بقیه جاها بوده است زیرا پایتخت سینماست. و این آن را از هر جای دیگری متمایز می کند. در واقع من فکر می کنم شرق و غرب آمریکا تفاوت چندانی با هم ندارند. آنجایی که تفاوت دارد، جنوب و بخش مرکزی کشور است و شما می توانید این را با نگاه کردن به نقشه های انتخابات که نشانگر جو سیاسی کشور است، دریابید. من فکر می کنم اروپایی ها احتمالاً نمی توانند درک کنند که آمریکا چقدر ناهمگن است. آن ها به آمریکا مثل کشوری همانند کشور خودشان فکر می کنند.

در مورد خود من، همان طور که گفتم خانواده پدرم در شرق آمریکا بودند و من در آنجا بستگانی داشتم که آن ها را خوب نمی شناختم. من چند بار قبلاً به آنجا رفته بودم و دوستی داشتم که او هم علاقه مند به زندگی در شرق آمریکا بود. بنابراین تصمیم گرفتیم که

این کار را با هم انجام دهیم. مثل یک ماجراجویی بود.

دی لایت: شما در شرکت میتر در بوستن کار کردید.

لیسکوف: برای یکسال. میتر شرکتی است که در اصل بر روی قراردادهای دولتی کار می کند. حجم کارهای پژوهشی که در این شرکت انجام می شود نیز قابل ملاحظه است و البته من در آن زمان، فقط به عنوان یک برنامه نویس در آنجا مشغول به کار بودم. و کد می نوشتم.

دی لایت: به چه زبانی برنامه نویسی می کردید؟

لیسکوف: فرترن. در اولین روزی که کارم آغاز شد، یک جزوه راهنمای فرترن به من دادند و چند مسئله کوچک که برای آن ها برنامه بنویسم. بدین ترتیب من همه چیز را پیش خود یاد گرفتم.

دی لایت: پس از آن برای ادامه تحصیل به استنفورد رفتید، درست است؟

لیسکوف: در واقع من یکسال در شرکت میتر کار کردم و یکسال هم در دانشگاه هاروارد. در هاروارد، من به عنوان برنامه نویس بر روی پروژه ترجمه زبان کار کردم. در آنجا به زبان همگذاری (اسمبلی) برنامه می نوشتم و از یک برنامه بزرگ به زبان همگذاری نیز نگهداری می کردم. این تغییر شغلم به خاطر این بود که اولاً تجربه جدیدی برای من بود و ثانیاً من در کمبریج زندگی می کردم و رفت و آمد به هاروارد برایم خیلی راحت تر بود. آن ها بر روی زبان انگلیسی کار می کردند و برنامه ای که نگهداریش به من محول شده بود، تجزیه جملات انگلیسی بود. البته هدف پروژه، ترجمه بین زبان های مختلف بود.

دی لایت: نامی که الان به ذهنم می آید چامسکی است. اما البته این به معنی آن که شما با او کار می کردید نیست.

لیسکوف: نه، چامسکی در هاروارد نبود. من با تونی اوتینگر کار می کردم و فکر می کنم سوسومو کونو پژوهشگری بود که با او کار می کرد. من اوتینگر را خوب نمی شناختم و با او تعامل زیادی نداشتم زیرا من فقط یک برنامه نویس بودم و او مدیر پروژه بود. تعامل من بیشتر با سوسومو بود که از آن هنگام تا کنون دیگر او را ندیده ام. اما یادتان باشد که من حتی دانشجوی تحصیلات تکمیلی هم نبودم، در حالی که آن ها عضو هیئت علمی بودند. بنابراین، فاصله زیادی بین ما وجود داشت.

دی لایت: شما پس از آن به استنفورد رفتید که در آن زمان هنوز دانشکده دانش رایانه (علوم کامپیوتر) نداشت. آیا در آنجا ویرث، کنوت یا فلویید را دیدید؟

لیسکوف: هنگامی که من در سال ۱۹۶۳ به آنجا رفتم، هنوز هیچیک از آن ها آنجا نبودند. جان مک کارتی آنجا بود و من کارم را با او آغاز کردم. نیکلاوس ویرث، دقیقاً مطمئن نیستم، شاید سال ۱۹۶۵ به آنجا آمد. کنوت هم پس از آن که من آنجا را ترک کردم آمد. شاید به عنوان بازدیدکننده در آنجا بوده اما من واقعاً

6- maintenance

7- parse

چیزی که او می‌خواست این بود که به کتاب‌های شطرنج مراجعه شود و آنچه کتاب‌ها می‌گویند، تبدیل به کد شود. در آن روزها شطرنج، چالش بزرگی بود. البته امروز دیگر چنین نیست و افراد از جستجوهای عمیق برای یافتن راه‌حل استفاده می‌کنند. اما آن روزها چنین جستجوهای در اختیار نبود و شما باید هوشمندی بیشتری برای هرس کردن^۸ درخت بازی به خرج می‌دادید تا بتوانید بدون جستجوی زیاد به جواب‌های مؤثر و موفقیت‌آمیزی برسید.

دی‌لایت: شما به چه زبانی برنامه‌نویسی می‌کردید؟
لیسکوف: لیسپ. و باید آن را یاد می‌گرفتم. دارم سعی می‌کنم که به یاد آورم آیا پاسکال را هم یاد گرفتم؟ مطمئن نیستم که پاسکال در آن موقع هنوز عرضه شده بود یا نه. من یک زبان دیگر را هم یاد گرفتم، شاید الگول. بله مطمئنم الگول بود. این که واقعاً به این زبان برنامه‌نویسی کردم یا نه، بحث دیگری است. فکر می‌کنم در بعضی از درس‌ها که از ماشین باروز ۵۰۰۰ استفاده می‌کردیم به زبان الگول برنامه می‌نوشتیم.

دی‌لایت: با وجودی که هر کس برای خودش مستقل کار می‌کرد، آیا شما با مک‌کارتی زیاد صحبت می‌کردید؟

لیسکوف: نه، هر از گاهی او را می‌دیدم. وقت ملاقات گرفتن از او خیلی سخت بود. و ملاقات‌هایمان هم خیلی کوتاه و مختصر بود.
دی‌لایت: آیا با وجود این، چیزی درباره کارهای او بر روی الگول به گوشتان خورد؟

لیسکوف: نه، من اطلاعات کمی درباره کارهای او بر روی پروژه‌های دیگر داشتم. من بر روی کار خودم متمرکز بودم و فکر می‌کنم راه موفقیت این است که فرد دیدگاهش را بر روی موضوع مورد نظر کاملاً باریک و محدود کند و اجازه ندهد چیزهای دیگر حواسش را پرت کنند. این روش کار کردن من است. من در آزمایشگاه بودم و راجردی بر روی بازوی روبات که موضوع پایان‌نامه دکتریش بود کار می‌کرد. و فرد دیگری به نام جیم پینتر هم بود که بر روی نظریه محاسبات با مک‌کارتی کار می‌کرد و موضوع پایان‌نامه‌اش برای من جالب‌تر از موضوع پایان‌نامه خودم بود چون من در اصل یک آدم سیستمی هستم نه هوش مصنوعی. من درس‌های زیادی با دانا اسکات که در آن زمان در استنفورد بود گرفتم. بدین ترتیب بود که مطالب مربوط به منطق را فرا گرفتم، و البته هیچ‌گاه هیچ کاری با آن انجام ندادم. (خنده لیسکوف)

دی‌لایت: آیا درباره ماشین تورینگ هم از اسکات چیزی یاد گرفتید؟

لیسکوف: یاد نمی‌آید. آنچه یادم می‌آید، تصمیم‌ناپذیری^۹ و مطالب مربوط به آن بود.

به یاد نمی‌آورم زیرا آزمایشگاه هوش مصنوعی که بیشتر کار من در آنجا بود، خارج از پردیس اصلی دانشگاه قرار داشت، در بالای تپه. زمانی که من دانشجوی سال سوم دوره دکترای بودم و بیشتر کارم در امر پژوهش بود، تقریباً هرگز به ساختمان‌های مرکزی دانشگاه رفت و آمد نمی‌کردم. به همین خاطر، با آدم‌ها زیاد تعامل نداشتم. احتمالاً در سال دوم بود که درسی با نیکلاوس ویرث داشتم. بنابراین، حتماً هنوز در پردیس اصلی بودم وقتی که او به استنفورد آمد.

دی‌لایت: ظاهراً شما همان روز نخست که به استنفورد رفتید، مک‌کارتی را دیدید [۱۱۴]. درست است؟

لیسکوف: من این‌گونه به یاد می‌آورم اما مطمئن نیستم که حتماً همین‌طور بوده باشد. ما هر دو از پله‌ها بالا می‌رفتیم و من از او پرسیدم که می‌تواند راهنمای من شود؟ اکنون که به گذشته می‌نگرم، این داستان خیلی غیرمحتمل به نظر می‌رسد. بنابراین واقعاً نمی‌دانم چه اتفاقی افتاد. اما آنچه مسلم است این که من او را خیلی زود دیدم چون کارم با او را تقریباً بلافاصله شروع کردم.
دی‌لایت: آیا پیش از آن که به آنجا بروید می‌دانستید او کیست؟
لیسکوف: نه.

دی‌لایت: شما شخصیت او را در آن زمان چگونه توصیف می‌کنید و پرسش دیگرم این است که خود شما آدم کمروبی بودید یا آن که از همان موقع خیلی به خودتان اطمینان داشتید و می‌دانستید در چه مسیری گام برمی‌دارید؟

لیسکوف: نه، نه. من خیلی خجالتی بودم و اعتماد به نفس پائینی داشتم. اما در مورد مک‌کارتی، فاصله زیادی بین خودم و او حس می‌کردم. او به‌عنوان استاد راهنما، خیلی در دسترس نبود. از آن آدم‌هایی نبود که با دانشجویانش به‌طور مرتب تعامل داشته باشد و یا ترتیباتی فراهم کند که دانشجویان درباره کارهایشان با هم صحبت کنند و از هم ایده بگیرند. هر کس به‌طور مستقل بر روی پروژه خودش کار می‌کرد.

دی‌لایت: آیا این رفتار در آن زمان عادی بود یا فقط جان مک‌کارتی این چنین بود؟

لیسکوف: من فکر می‌کنم این بیشتر به خود استاد راهنما برمی‌گردد. هر استادی سبک خودش را دارد.

دی‌لایت: پایان‌نامه دکترای شما درباره پایان بازی‌های شطرنج بود. می‌شود کمی درباره آن توضیح دهید؟

لیسکوف: پایان‌نامه من درباره پایان بازی‌های قابل حل بود. مثلاً شاه و وزیر در مقابل شاه، یک پایان بازی قابل حل است. هدف پایان‌نامه، یافتن راهبردهایی بود که برنامه را قادر سازد تا این بازی‌ها را همیشه با برد به پایان رساند. جان مک‌کارتی فکر می‌کرد من فرد مناسبی برای انجام این پایان‌نامه هستم زیرا من شطرنج بازی نمی‌کردم و او حس می‌کرد اگر کسی شطرنج‌باز باشد، ایده‌های خودش را برای انجام کار دارد. در حالی که

8- prune

9- undecidability

کارهای اولیه هم‌عصران

دی‌لایت: من در اینجا کتاب «بررسی خودکارها»^{۱۰} [۲۷۲] نوشته شانون و مک‌کارتی در سال ۱۹۵۶ را دارم که کاملاً نظری است.

لیسکوف: فکر می‌کنم هیچگاه درسی در نظریه دانش رایانه (علوم کامپیوتر) نگرفته باشم. یک درس درباره برنامه‌های مترجم (کامپایلر) داشتم. یک درس هم درباره آنالیز عددی، چون به آن نیاز داشتم. البته چند درس هم با دانا اسکات گرفتم اما آن‌ها بیشتر در زمینه منطق بودند تا دانش رایانه. بنابراین، این کتاب اصلاً به نظرم آشنا نمی‌آید.

دی‌لایت: عنوان مقاله مک‌کارتی در این کتاب «وارون‌سازی»^{۱۱} توابع تعریف شده توسط ماشین‌های تورینگ است.

لیسکوف: نه، اصلاً یاد نمی‌آید.

دی‌لایت: او به کمک ماشین‌های تورینگ توضیح می‌دهد که چگونه می‌توان مسائل فکری را حل کرد. شما بر روی پایان بازی‌های شطرنج کار کرده‌اید. بنابراین من فکر کردم شاید ارتباطی با کار شما داشته باشد.

لیسکوف: در پایان‌نامه من، هیچ چیز صوری^{۱۲} وجود ندارد. جان مک‌کارتی با دانا اسکات هم کار می‌کرد، احتمالاً چون به تحلیل‌های صوری علاقه‌مند بود. اما من دقیقاً نمی‌دانم که آن‌ها چه کار می‌کردند.

دی‌لایت: من کتاب «برنامه‌نویسی رایانه و سیستم‌های صوری»^{۱۳} [۲۷] نوشته برافورت و هرشبرگ در سال ۱۹۶۱ را نیز همراه آورده‌ام. این کتاب، نتیجه کارهای مشترک چند منطق‌دان و برنامه‌نویس در شمال هلند است. مک‌کارتی هم در این کتاب، مقاله‌ای دارد با عنوان «مبنایی برای نظریه ریاضی محاسبات». او در این مقاله به «پندگیر»^{۱۴} اشاره کرده است. (پندگیر نام یک برنامه رایانه‌ای فرضی است که توسط جان مک‌کارتی ارائه شده و احتمالاً نخستین پیشنهادی است که برای به‌کارگیری منطق برای نمایش اطلاعات در رایانه به عمل آمده است. مترجم)

لیسکوف: من چند تا از مقاله‌های مک‌کارتی را خوانده‌ام. مقاله حساب‌گزاره‌ها^{۱۵} را یاد می‌آید اما این مقاله‌ای که شما می‌گویید را نه. پندگیر یک برنامه هوش مصنوعی بود. فکر می‌کنم مقاله‌ای که درباره پندگیر بود توسط نیوول و سایمون نوشته شده بود. بنابراین احتمالاً از گروه کارنگی ملون بیرون آمده باشد.

دی‌لایت: شما در سال ۱۹۷۰ ازدواج کردید و به شرکت میتر بازگشتید، اما این بار به‌عنوان پژوهشگر. تصوّر این است که در آن زمان دیگر پی برده بودید که واقعاً به دنبال چه چیزی هستید.

لیسکوف: من هدفم را پیش از آن مشخص کرده بودم. سال سوم

یا چهارم دوره دکتری، من به سیستم‌ها بیشتر از هوش مصنوعی علاقه‌مند شده بودم. یک علتش این بود که حس می‌کردم هوش مصنوعی خیلی سخت است. من به یادگیری ماشینی^{۱۵} خیلی علاقه‌مند شده بودم. اما در آن زمان، یادگیری ماشینی تماماً بر پایه این ایده بود که ماشین‌ها باید همان کارهایی را که آدم‌ها می‌کنند، انجام دهند. البته امروز یادگیری ماشینی به هیچوجه شبیه یادگیری انسان‌ها نیست. آن‌ها از آمار و نظیر آن استفاده می‌کنند. من حس می‌کردم که در چیزهایی که برایم جالب بودند پیشرفت نمی‌کردم. به این جهت، این موضوع را برای پایان‌نامه‌ام انتخاب کردم که می‌توانستم سخت روی آن کار کنم و آن را به پایان ببرم. موضوع پایان‌نامه من به خودی خود جالب بود اما برای من گیرایی خاصی نداشت. در این بین، من به موضوع زبان‌ها و سیستم‌ها خیلی علاقه‌مند شده بودم و می‌دانستم که در آینده موضوع پژوهشی‌ام را تغییر خواهم داد. بعد تصمیم گرفتم که این کار را به بعد از تکمیل پایان‌نامه دکتری‌ام موکول کنم. بدین ترتیب، به محض آن که به شرکت میتر بازگشتم، این کار را انجام دادم و شروع به کار بر روی سیستم‌ها کردم.

دی‌لایت: در آنجا بود که شما یک معماری رایانه طراحی کردید که به سیستم عامل ونوس انجامید. شما همچنین در زمینه روشگان^{۱۶} برنامه‌نویسی خیلی فعال شدید. درست است؟

لیسکوف: بله، اما کار من بر روی روشگان برنامه‌نویسی در ۱۹۷۰ یا نزدیک به آن آغاز شد. در نخستین سال‌های حضورم در میتر، بر روی ساخت ماشین ونوس بر روی اینتردیتا^{۱۷} ۳ و سپس سیستم عامل ونوس با استفاده از ماشین ونوس کار می‌کردم. تنها پس از پایان این پروژه بود که کار بر روی روشگان برنامه‌نویسی را آغاز کردم.

دی‌لایت: دهه ۱۹۶۰، سال‌های گول ۶۰ و گول ۶۸ نیز بود. آیا این زبان‌ها تأثیر مستقیمی بر روی شما داشتند؟

لیسکوف: گول ۶۸ مطمئناً تأثیرگذار بود. اما احتمالاً تا وقتی که به ام‌آی‌تی آمدم آن را خوب نمی‌شناختم. هنگامی که در میتر بودم، تمرکز بر روی کارهای سیستمی بود. وقتی که به ام‌آی‌تی آمدم، شروع به بررسی زبان‌ها کردم.

دی‌لایت: من در اینجا مجموعه مقالات همایش «مهندسی نرم‌افزار» در گارمیش ۱۹۶۸ را دارم [۲۲۱]. عبارت «بحران نرم‌افزار» در آنجا و به همراه مقاله دایکسترا با عنوان «کنترل پیچیدگی از طریق رده‌بندی پایگانی»^{۱۷} تابع و درجه تغییرپذیری مطرح شد.

لیسکوف: من آن مقاله خاص را به یاد نمی‌آورم. دایکسترا مقالات زیادی نوشته است و بجز آن‌هایی که در مقالات خودم به آن‌ها ارجاع داده‌ام، نمی‌توانم بگویم که بقیه مقالات او را هم خوانده‌ام یا نه.

15- machine learning

16- methodology

17- hierarchical

10- Automata Studies

11- inversion

12- formal

13- Advice Taker

14- propositional calculus

دی لایت: در مورد رویکردهای داده محور^{۲۳} شما در مقاله سال ۱۹۷۵ خود با عنوان «نوع‌های داده‌ها^{۲۴} و سخت برنامه» [۱۷۳] به کار جورج میلی [۱۹۱]، جی ارلی [۸۸] و رابرت بالزر [۱۲] اشاره کرده‌اید.

لیسکوف: من به‌طور مبهمی جی ارلی را به یاد می‌آورم. او بر روی زبان‌های برنامه‌نویسی کار می‌کرد. احتمالاً هنگامی که بر روی نوع داده‌های انتزاعی، پس از پیوستن به ام‌آی‌تی، کار می‌کردم، مقاله‌اش را خوانده‌ام.

دی لایت: جی ارلی طرفدار جداسازی مشخصات^{۲۵} از پیاده‌سازی بود. جورج میلی طرفدار نوع داده تعریف شده توسط کاربر بود. او گفته است:

«چیزی که واقعاً نیاز داریم، نوع داده‌های کمتر، همراه با سازوکاری است که به کاربر اجازه دهد نوع داده‌های خودش را تعریف کند.» [۱۹۱]

لیسکوف: من مقاله‌های جورج میلی را خوانده‌ام اما در آن‌ها معمولاً به کارهای دیگران اشاره نشده است. جمله‌ای که از او نقل کردید، تقاضای آشکاری برای نوع داده‌های انتزاعی است.

دی لایت: رابرت بالزر چی؟ آیا در آن زمان او را می‌شناختید؟ عنوان دست‌نویس سال ۱۹۶۷ او «برنامه‌نویسی بدون داده^{۲۶}» [۱۲] بود.

لیسکوف: پایان‌نامه دکتریش بود. آن را به یاد می‌آورم. من او را همان اوایل ملاقات کردم، فکر می‌کنم اوایل دهه ۱۹۷۰. پایان‌نامه دکتریش او یکی از مقالات متعددی بود که پس از ۱۹۷۲ خواندم. با استیو زیلس آن را خواندیم. مقاله جی ارلی را نیز با هم خواندیم - حالا یادم آمد. احتمالاً مقاله جورج میلی را زودتر خوانده بودم زیرا بیشتر در حوزه روشگان برنامه‌نویسی بود.

چیزی که رابرت بالزر در پایان‌نامه دکتریش پیشنهاد می‌کند، یک شیوه غیرطبیعی نگاه کردن به داده‌ها، برحسب مجموعه ثابتی از عملیات بدون هرگونه انتزاع است. بنابراین، پنهان‌سازی اطلاعات^{۲۷} وجود ندارد. او در مورد قواعد نحوی^{۲۸} نیز شگردهایی را به کار برده است به نحوی که عملیات را می‌شد به نحو خاصی نوشت. کار او گامی به جلو بود اما وارد حوزه انتزاع داده‌ها نمی‌شد. این نکته را در نظر داشته باشید که من برای آماده کردن سخنرانی دریافت جایزه تورینگ، به عقب برگشتم و بسیاری از مقالات را دوباره خواندم. بنابراین آنچه امروز به شما می‌گویم، آن چیزی نیست که از خواندن این مقاله در سال ۱۹۷۲ یادمانده زیرا سال گذشته دوباره آن را خوانده‌ام.

دی لایت: در یک تصویر پژوهشی بزرگ‌تر از زبان‌های توسعه‌پذیر (سطح بالا)، کار بالزر در کجا قرار می‌گیرد؟ چرا در آن زمان،

دی لایت: یکسال پس از همایش گارمیش ۱۹۶۸، همایش دیگری برگزار شد. مجموعه مقالات این همایش [۳۳] توسط باکستون و رندل ویرایش شده است. مقاله دایکسترا در این مجموعه، «برنامه‌نویسی ساخت‌یافته» بود که البته نباید آن را با مقاله «یادداشت‌هایی درباره برنامه‌نویسی ساخت‌یافته» که در کتاب معروف «برنامه‌نویسی ساخت‌یافته» [۴۵] آمده، اشتباه گرفت.

لیسکوف: آن کتاب را من به دقت خوانده‌ام. مطمئناً آثار دیگری از دایکسترا را هم خوانده‌ام اما اکنون به یاد نمی‌آورم.

دی لایت: مقاله «برنامه‌نویسی ساخت‌یافته» دایکسترا درباره یک گردنبند مروارید است. او از مرواریدها برای تشبیه استفاده کرده است.

لیسکوف: بله، من آن مرواریدها را به یاد می‌آورم. اگر درباره‌اش فکر کنید، تشبیه غریبی است. زیرا واقعاً شبیه گردنبند نیست. در مورد ایده طراحی بالا به پائین، کاملاً مطمئنم که مقاله ویرث درباره پالایش گام‌به‌گام^{۱۸} بود که به نظر من از همه تأثیرگذارتر بود. البته همه آن‌ها درباره یک چیز حرف می‌زدند و دایکسترا هم مقاله‌ای نوشته است که آن نیز در همین باره است.

دی لایت: نظرتان در مورد این یکی مقاله دایکسترا چیست: «ساختار سیستم چند برنامه‌ای^{۱۹} THE»؟ [۷۰]

لیسکوف: بله، من آن را خوانده‌ام. این مقاله‌ای است که دایکسترا در آن نشانبرها^{۲۰} را معرفی کرده است. من در ماشین ونوس از نشانبرها استفاده کردم و به این نتیجه رسیدم که آن‌ها خیلی جالب هستند و برای سخت‌افزار خیلی خوب است که پشتیبانی مستقیمی برای همگام‌سازی^{۲۱} داشته باشد. در واقع، هنگامی که به این موضوع در آن زمان فکر می‌کنید، کار جالبی بود. امروز دستور test-and-set یا چیزی شبیه آن با پشتیبانی اولیه در سطح سخت‌افزار وجود دارد اما در آن روزها، این چیزها در اختیار نبود و شاید علتش این بود که کسی به عملکرد چند برنامه‌ای فکر نمی‌کرد. من دایکسترا را ملاقات کرده‌ام. فکر می‌کنم او به شرکت میتر آمد و من با او صحبت کردم. دقیقاً سالش را به یاد نمی‌آورم اما احتمالاً ۱۹۶۹ بود و ما با هم درباره ایده‌های او در آن مقاله صحبت کردیم. پیش از آن بود که من کار بر روی سخت‌افزار ونوس را تمام کنم.

دی لایت: دایکسترا را چه جور آدمی توصیف می‌کنید؟ فکر می‌کنم لهجه هلندی داشت.

لیسکوف: اما انگلیسی‌اش خیلی خوب بود. چیزی که به یاد می‌آورم این است که با هم صحبتی درباره ساخت‌های ابتدایی^{۲۲} همگام‌سازی داشتیم.

23- data-centric

24- data types

25- specification

26-Dataless Programming

27- information hiding

28- syntax

18- stepwise refinement

19- multiprocessing

20- semaphores

21- synchronization

22- primitives

زبان‌های عمومی^{۲۹} این قدر مورد بحث قرار می‌گرفتند؟

لیسکوف: خوب، امروز می‌دانیم که هر برنامه‌ای را می‌توانیم به یک زبان عمومی مانند C یا جاوا بنویسیم، اما در آن زمان این نگرانی وجود داشت که آیا می‌شود این کار را کرد یا نه. منشأ این نگرانی این بود که هرگاه یک زبان سطح بالاتر به وجود می‌آوردید، از ویژگی‌های ماشین زیرین می‌کاستید زیرا آن ماشین را با یک چیز سطح بالاتر جایگزین می‌کردید و مجبور می‌شدید از آن ماشین فقط از طریق آن چیز سطح بالاتر استفاده کنید. و آن‌ها نگران بودند که از دست دادن قابلیت بیان از سطح پائین‌تر به سطح بالاتر، مشکلات جدی به وجود آورد. و حتی امروز هم اگر شما بخواهید سیستم‌های سطح پایین را به جاوا بنویسید، مشکل‌زاست. شما بعضی چیزها را به C می‌نویسید تا با کارایی بیشتری انجام گیرد اما به‌طور کلی، امروز می‌دانیم که برنامه‌ها را می‌توان به هر زبان برنامه‌نویسی عمومی نوشت. کار بر روی زبان‌های توسعه‌پذیر، با هدف پرداختن به این مسئله با زبان‌های عمومی، صورت می‌گرفت. همان‌گونه که شومان و جوراند در مقاله فوق‌العاده جالبشان [۲۷۱] گفته‌اند، دو رویکرد متفاوت وجود داشت: نحوی و معنایی^{۳۰}.

منظور از نحوی این بود که شما می‌توانستید قواعد نحوی زبان برنامه‌نویسی را توسعه دهید. از نظر ظاهری، این ایدهٔ موجهی است. شما پیش خود می‌گوئید «خوب، من از فاکتوریل زیاد استفاده می‌کنم. پس چرا نتوانم علامت تعجب (!) را بگیرم و به آن معنی تابع فاکتوریل بدهم؟ این کار، نوشتن برنامه را برای من بسیار راحت‌تر خواهد ساخت.» اما در واقع، این ایدهٔ بسیار بدی است. زیرا می‌دانیم که امروز، خواندن برنامه‌ها بسیار مهم‌تر از نوشتن آن‌هاست. برنامه، یکبار نوشته می‌شود، توسط نویسنده‌اش بارها و بارها خوانده می‌شود و بعد از آن که ده بار آن را خواندید، فراموش می‌کنید که چکار می‌کردید و بعد توسط دیگران در گروه شما خوانده می‌شود و پنج سال بعد، یکنفر دیگر مسئول نگهداری آن می‌شود. بنابراین، خواندن برنامه‌ها آن چیزی است که اهمیت دارد نه نوشتن آن‌ها. برای خواندن برنامه، باید قواعد نحوی آن را درک کنید و در نتیجه، چیزی که می‌خواهید این است که قواعد نحوی تغییر نکنند.

نکتهٔ دیگری که شومان و جوراند درباره‌اش صحبت کرده‌اند، چیزی است که از آن به‌عنوان توسعهٔ معنایی نام برده‌اند. پایان‌نامهٔ رابرت بالزر در ۱۹۶۷ مثالی از این است. او فرض کرده بود که تمام نوع داده‌ها، مجموعه هستند. او گفته است که هر نوع داده باید چهار عمل داشته باشد. یک عمل برای «قرار دادن» یک چیز در مجموعه، یکی برای «حذف» یک چیز از مجموعه، یکی برای «دیدن» این که چه چیزی در مجموعه هست و بالاخره یک عمل هم برای «اصلاح» یک چیز در مجموعه. و او سپس روش‌هایی برای این کار با استفاده از قواعد نحوی خاص ابداع کرده است. بنابراین،

ملاحظه می‌کنید که این یک گام به جلو در مسیر نوع داده‌های انتزاعی است، البته بجز این که تمام داده‌ها مجموعه نیستند و خیلی دردرس‌ساز می‌شود اگر بگوئید همهٔ آن‌ها باید یک مجموعه از عملیات را دارا باشند. نه تنها این، بلکه درون‌بندی^{۳۱} نیز وجود نداشت. بنابراین، فقط مثل اضافه کردن چاشنی به زبان بود و هیچ راهی برای حفاظت از کاری که انجام می‌دادید، فراهم نمی‌کرد.

دی‌لایت: بالزر می‌گفت برنامه‌نویس برای نوشتن برنامه‌ای برای یک مسئله، باید چهار گام بردارد:

۱- تعریف مسئله

۲- تحلیل نیازهای پردازشی

۳- انتخاب نحوهٔ نمایش داده‌ها براساس آن نیازها

۴- نوشتن برنامه

او می‌خواست گام‌های سوم و چهارم را جابجا کند زیرا:

«تغییر نحوهٔ نمایش داده‌ها معمولاً مستلزم تغییرات گسترده در کد نوشته شده است.» [۱۲]

برنامه‌نویسی بدون داده، زبانی بود که او برای انجام این کار پیشنهاد کرد.

لیسکوف: می‌توان مشاهده کرد که داشتن قواعد نحوی ثابت برای تعامل با ساختمان داده‌ها از این ایده پشتیبانی می‌کند زیرا شما می‌توانید آزادانه ساختمان داده‌های زیرین را تغییر دهید. اما در روش‌شناسی او چیزی وجود نداشت که تضمین کند کدی که از ساختمان داده‌ها استفاده می‌کند، از نحوهٔ نمایش استفاده نمی‌کند. بنابراین، وارد انتزاع داده‌ها نمی‌شد.

هنگامی که دربارهٔ نطقم در مراسم جایزهٔ تورینگ فکر می‌کردم، یکی از چیزهایی که به نظرم آمد این بود که چگونه می‌توان انتزاع داده‌ها را همه جا دید، در حالی که به‌طور کامل تعریف نشده باشد. میلی و بالزر آن را پیشنهاد کرده‌اند اما انتزاع داده‌ها در کارهای دایکسترا و ویرت کمتر به چشم می‌آید. آن‌ها در مورد انتزاع داده‌ها فکر نمی‌کردند و در عوض به فکر فرایند برنامه‌نویسی بودند.

دی‌لایت: بنابراین نادرست است اگر فرض کنیم که شما در اوایل دههٔ ۱۹۷۰ از این کار آگاهی داشتید؟

لیسکوف: در اوایل دههٔ ۱۹۷۰، من از پژوهش‌های روشنگان برنامه‌نویسی آگاهی داشتم. من در آن هنگام پایان‌نامهٔ بالزر را نخوانده بودم زیرا پایان‌نامه‌اش دربارهٔ سازوکار زبان برنامه‌نویسی بود. زمانی که من کار بر روی روشنگان برنامه‌نویسی را آغاز کردم، و پیش از آن که کار بر روی نوع داده‌های انتزاعی را آغاز کنم، از کارهای پاراناس، دایکسترا، ویرت و بعضی مقالات میلی آگاهی داشتم.

دی‌لایت: آیا روش SODAS را که پاراناس در همایش رایانه در پائیز ۱۹۶۷ ارائه کرد، به یاد می‌آورید؟ SODAS سرنام «توصیف و شبیه‌سازی با گرایش ساخت‌یافته»^{۳۲} است [۲۳۶]. او در این مقاله

31- encapsulation

32- Structured Oriented Description And Simulation

29- universal languages

30- semantic

جدول ۶-۱: مقایسه بین سیستم عامل‌های THE و ونوس
(منبع: کتاب سیلبرشائز و گالوین)

THE	ونوس
برنامه‌های کاربر میانگیری ۱ برای دستگاه ورودی/خروجی	برنامه‌های کاربر
گرداننده‌های دستگاه	گرداننده‌های دستگاه و زمانبندها
مدیریت حافظه	حافظه مجازی مجرای ورودی/خروجی
زمانبندی پردازنده مرکزی	زمانبندی پردازنده مرکزی مفتر دستورالعمل
سخت‌افزار	سخت‌افزار

که بعضی‌ها پیشنهاد می‌کردند و در سطح N شما فقط می‌توانستید از سطح N-1 استفاده کنید، نیست.

دی‌لایت: آیا به یاد می‌آورید که چرا به فکر دور زدن سطح‌ها افتادید؟

لیسکوف: اگر این کار را نکنید، چیزی که در هر سطح به دست می‌آورد، چیزی است که باید دستیابی به تمام کارکردهای سطح پایین‌تر را فراهم سازد و این فقط باعث افزودن پیچیدگی بیشتر به طراحی می‌گردد. روشی که من امروز به آن فکر می‌کنم این است که به هنگام طراحی، مجموعه‌ای از انتزاع‌ها را در دسترس داشته باشید. دلیلی وجود ندارد که اگر از آن مجموعه از انتزاع‌ها استفاده کردید، مجبور باشید همه آن‌ها را در یک آبرانتزاع^{۳۴} که شامل همه آن‌ها باشد، بسته‌بندی کنید. بهتر است به مجموعه‌ای فکر کنید که با آن می‌توانید کار کنید. این که برخی از انتزاع‌ها که از آن استفاده می‌کنید ممکن است از برخی انتزاع‌های دیگر استفاده کنند، اهمیتی ندارد.

دی‌لایت: در سال ۱۹۷۱ در بلفاست، تونی هور و ران پروت همایشی را درباره سیستم‌های عامل ترتیب دادند [۱۳۴]. شما در آن شرکت نکردید، دایکسترا هم همین‌طور. هنگامی که مجموعه مقالات آن همایش را مطالعه می‌کردم، متوجه سه چیز اساسی شدم. نخست آن که در طول آن همایش، همه آن‌ها درباره سیستم عامل THE دایکسترا حرف می‌زدند. دوم آن که هور نسبت به لایه‌های ماشین‌های مجازی تردید داشت. و سوم آن که هور، رندل و چند نفر دیگر دیدگاه‌های متفاوتی درباره این که سیستم عامل واقعاً چیست یا چه باید باشد، داشتند. برای نمونه، هور گفته است: «منظور اصلی سیستم عامل رایانه، به اشتراک گذاشتن سخت‌افزاری که تحت کنترل دارد، در بین تعدادی کاربر که تقاضاهای غیرقابل پیش‌بینی برای در اختیار گرفتن منابع دارند است و هدفش انجام دادن این کار به نحوی قابل اطمینان، کارآمد و بدون سربرار می‌باشد.» [۱۳۴]

آنچه را رویکرد بالا به پایین به طراحی سیستم نامیده، توصیف کرده است. به گفته خود او:

«این روش مستلزم شروع از «بالا» با مشخصات کامل رفتار مطلوب سیستم، و سپس شکستن سیستم به مؤلفه‌های کوچک‌تر و کوچک‌تر تا جایی که سیستم بر حسب اجزاء پایه تعریف و مشخص گردد است. مشخصات در هر سطح باید ضوابط ارزیابی برای سطح پایین‌تر بعدی را فراهم سازد.» [۲۳۶]

چیزی که برای من جالب است این است که این شیوه کار کردن بالا به پایین با سطوح مختلف، ظاهراً از جاهای مختلفی سرچشمه گرفته است. فقط دایکسترا نبوده که این ایده را داشته است.

لیسکوف: بله، من عقیده دارم هنگامی که زمان جوانه‌زدن ایده‌ها باشد، غالباً به‌طور همزمان در جاهای مختلف می‌روید. دقت کنید که دایکسترا و پاراناس دو شخصیت کاملاً متفاوت بودند. دایکسترا نماینده مکتب اروپایی بود و پاراناس از مکتب آمریکایی می‌آمد. با وجود این، باز هم نقاط مشترکی وجود داشت.

سیستم عامل ونوس

دی‌لایت: من در اینجا یک نسخه از مقاله شما با عنوان «طراحی سیستم عامل ونوس» [۱۷۲] را دارم. شما به خاطر آن، برنده جایزه بهترین مقاله در سال ۱۹۷۲ شدید. علتش را به یاد می‌آورید؟

لیسکوف: نمی‌دانم. هرگز درباره این پرسش فکر نکرده‌ام. به احتمال بسیار زیاد به خاطر درهم آمیختن روشگان برنامه‌نویسی با طراحی سیستم عامل بوده است. من علاقه‌مند بودم بینم چگونه روشگان برنامه‌نویسی می‌تواند در طراحی ونوس به کار گرفته شود. این صحبت مربوط به بهار ۱۹۷۲ است. طراحی سیستم عامل در آن زمان یک موضوع پژوهشی بود. البته هنوز هم هست اما نه به آن صورت که در آن هنگام بود. در آن موقع افراد هنوز سعی می‌کردند بفهمند سیستم عامل چیست. سیستم عامل ونوس به عقیده من کمی غیرعادی بود. شما می‌توانید تأثیر سیستم عامل THE را در آن ببینید: ایده ساختن چیزها در لایه‌های مختلف. و نشانبرها، این که من آن‌ها را در سخت‌افزار قرار دادم، خیلی در این پروژه سودمند بود.

دی‌لایت: سیستم عامل THE دایکسترا و سیستم ونوس شما از نظر لایه لایه بودن، بی‌مانند و یگانه بودند. از نظر لایه‌بندی نیز شبیه به هم بودند. (جدول ۶-۱ را ببینید)

لیسکوف: این مفهوم سطوح انتزاع، چیزی است که من به آن اعتقاد ندارم. به عقیده من این چیزها بیشتر ساختاری شبیه گراف دارند. اما در آن زمان، این مفهوم از پایگان^{۳۳}، رایج بود و هر سطح بر روی سطح قبلی ساخته می‌شد. اگر شما به شکل ۲ در مقاله ونوس نگاه کنید، می‌بینید که با آن که من آن‌ها را سطح نامیده‌ام، اما از روی سطوح پریده‌ام. شبیه آن سطح‌بندی‌های دقیق و سخت‌گیرانه

34- super abstraction

33- hierarchy

بنابراین، اشتراک برای هور مسئله اصلی بود. اما رَندل می گفت که به اشتراک گذاری منابع، هدف اصلی سیستم عامل نیست:

«من حس می کنم که تأکید بر روی اشتراک در مقاله هور، صحیح نیست. اشتراک، راه حل است نه مسئله. توصیف من از سیستم عامل این است که چیزی است که اجازه می دهد (یا تلاش می کند که اجازه دهد) یک نفر به حداکثر کارایی و اطمینان پذیری^{۳۵} سخت افزار دست یابد. با بزرگ تر شدن و پرهزینه تر شدن سیستم، اشتراک یک روش واضح برای پرداختن به مسئله دستیابی به حداکثر کارایی است.» [۱۳۴]

به طور خلاصه، این پژوهشگران هنوز اتفاق نظر بر سر این که سیستم عامل چیست، نداشتند. هنگامی که شما در شرکت میتر کار می کردید نیز چنین چیزی را تجربه کرده بودید؟

لیسکوف: من دقیقاً به یاد نمی آورم و دیدگاه من در مورد سیستم های عامل تحت تأثیر چیزهایی است که امروز می دانم. در آن زمان، عملکرد چند برنامه ای شاید هنوز مسئله اصلی نبود و این می تواند دلیل خوبی باشد برای این که چرا آن ها این قدر درباره اشتراک صحبت می کردند. شما می توانید به سیستم عامل به صورت پشتیبان یک کاربر منفرد نگاه کنید و دستیابی به قدرت ماشین را در سطح انتزاعی تری فراهم کنید. و در واقع، اگر امروز به اجرای سیستم عامل بر روی رایانه شخصی خودتان فکر کنید، این همان اتفاقی است که دارد می افتد. در آن زمان، فکر می کنم خط فکری غالب درباره سیستم عامل این بود که ماشین ها گران قیمت بودند و شما می خواستید که کاربران هر چه بیشتری را به طور همزمان پشتیبانی کنید. و واضح بود که آن ها منابع را به اشتراک می گذاشتند و شما به دنبال پیدا کردن راهی بودید که به آن ها اجازه به اشتراک گذاری دهد بدون آن که تداخلی در کار یکدیگر ایجاد کنند و این کار را به نحو مؤثر و کارآمدی انجام دهد و امثال آن. ونوس یک سیستم چند برنامه ای بود. بنابراین، سیستمی بود که در اساس اجازه می داد برنامه های چند کاربر به طور همزمان بر روی یک سخت افزار مشابه اجرا گردند. همه چیز بر مبنای اشتراک بود. به همین دلیل بود که این انتزاع ها، این فرایندها یا ماشین های مجازی و این چیزهایی که من قطعه^{۳۶} نامیده بودم اما در اساس، پرونده بودند، وجود داشت. همه این ها، روش های انتزاعی کردن خدمات سخت افزار است به نحوی که انگار چند نمونه از آن ها وجود دارد. این روشی برای به اشتراک گذاری منابع موجود است.

دی لایت: هنگامی که شما این ماشین های مجازی را معرفی کردید، آیا تمایلی به حداقل رساندن تعداد آن ها داشتید؟ آیا معرفی ماشین های مجازی هزینه ای دربرداشت؟

لیسکوف: اگر درست به یادم بیاید، سیستم ونوس از ۱۶ کاربر همزمان پشتیبانی می کرد. معماری آن را به یاد نمی آورم که آیا چند فرایند اضافی و شناور داشت که سیستم عامل بتواند بعضی کارها را

35- reliability

36- segment

به طور موازی انجام دهد یا نه. من نمی دانم که داشتن ماشین های مجازی بیشتر، هزینه را بود یا نه اما انتقال^{۳۷} از یکی به دیگری، نوعاً پرهزینه بود. البته در زمانی که آن مقاله نوشته شد [۱۷۲]، درک زیادی نسبت به هزینه چنین انتقالی وجود نداشت.

دی لایت: در آن همایش سال ۱۹۷۱ در بلغاست، شرکت کنندگان درباره ساختن سیستم عامل در سطوح مختلف انتزاع صحبت می کردند. آن ها تردید داشتند که این روش درستی برای طراحی سیستم عامل باشد. رَندل به این موضوع اشاره کرده است که سیستم عامل THE دایکسترا، دچار کوبیدگی^{۳۸} بوده است (در سیستم حافظه مجازی به روش صفحه بندی^{۳۹}، به وضعیتی که بیشتر زمان پردازنده صرف آوردن و بردن صفحات بین حافظه اصلی و حافظه جانبی گردد، کوبیدگی می گویند. این وضعیت باعث کُند شدن سیستم می شود. مترجم):

«سیستم دایکسترا دچار کوبیدگی است. این مثالی است از این که سطوح انتزاعی سیستم THE کفایت نمی کند. من نمی خواهم علیه سطوح انتزاع موضع بگیرم اما مسئله کوبیدگی، مثال خیلی خوبی است برای این که سطوح انتزاع، حتی اگر به همان زیبایی که دایکسترا تعریف کرده هم تعریف شده باشند باز هم ناکافی هستند.» [۱۳۴]

به همین ترتیب، هور و برینچ هَنسن هم گفته اند که انتزاع ها نوعاً برای زمان بندی کارایی ندارند.

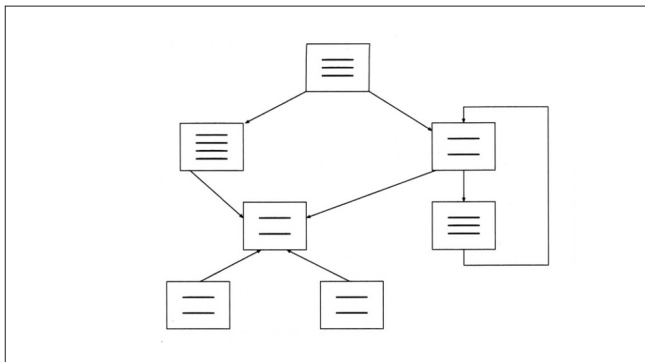
لیسکوف: زمان بندی، مسئله دشواری است. چند گروه پژوهشی مختلف در دهه ۱۹۷۰ بر روی آن کار کردند. من فکر نمی کنم که تقابل بین انتزاع و زمان بندی باشد. به نظر من، بیشتر به درک چگونگی پیاده سازی صحیح چنین کارکردی برمی گردد. پژوهش های مفصلی در این زمینه وجود دارند. از کجا می فهمید که این فرایند باید به آن یکی ترجیح داده شود و چگونه تصمیم می گیرید که کدام واحدها استفاده شوند؟ چنین پرسش هایی مورد مطالعه دقیق قرار گرفته اند. آن طور که به یاد می آورم، ونوس دارای سیستم زمان بندی پیچیده ای نبود. من نمی دانم که بر حسب زمان هم وقفه^{۴۰} می دادیم یا نه. فکر می کنم هرگاه یک فرایند نمی توانست هیچ کاری انجام دهد، پردازنده را به فرایند دیگری می دادیم. البته این در یک محیط بزرگ تر که منابع محدود هستند، کار نمی کند. و باید صادقانه بگویم که فکر نمی کنم کوبیدگی ارتباطی به این موضوع داشته باشد. نخست آن که اگر سعی کنید برنامه خود را اجرا کنید و برنامه شما به حافظه ای بیش از آنچه در اختیار دارد نیاز داشته باشد، کوبیدگی پیش می آید. این ارتباطی به انتزاع ندارد، یک واقعیت است، اما احتمالاً در آن زمان به خوبی درک نشده بود. دوم این که شما می توانید روش های پیچیده ای برای زمان بندی

37- switching

38- thrashing

39- paging

40- interrupt



شکل ۲- گراف وابستگی پودمان‌ها

معمولاً خیلی کامل نیست. و نیز استدلال صوری، با وجودی که پیشرفت‌های بسیاری ظرف ده سال گذشته با اثبات‌کننده‌های قضایا^{۴۴} به عمل آمده، هنوز خیلی دشوار است. آن‌ها به مشخصات صوری نیاز دارند و به دست آوردن مشخصات صوری خوب نیز آسان نیست.

دی‌لایت: شما در مقاله‌ی روشگان طراحی درباره‌ی سطوح انتزاع و برنامه‌نویسی ساخت‌یافته بحث کرده‌اید. برای سطوح انتزاع به دو قاعده اشاره کرده‌اید:

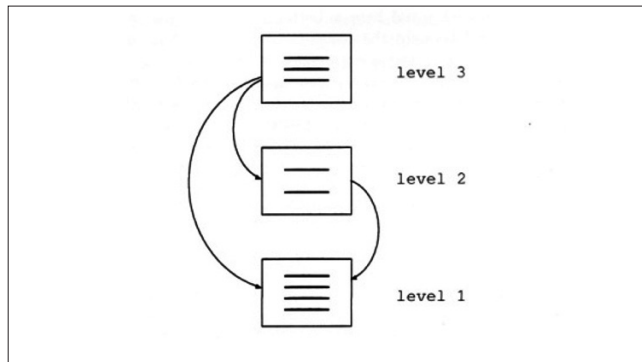
«هر سطح از انتزاع از گروهی تابع وابسته به هم تشکیل شده است. دو قاعده مهم بر سطوح انتزاع حاکم است. نخستین قاعده مربوط به منابع (دستگاه‌های ورودی/خروجی، داده‌ها) است: هر سطح دارای منابعی است که به‌طور انحصاری در اختیار دارد و سایر سطوح اجازه دسترسی به آن‌ها را ندارند. قاعده دوم در باره پایگان است:

سطوح پایین‌تر از وجود سطوح بالاتر آگاهی ندارند.» [۱۷۱]

در مورد برنامه‌نویسی ساخت‌یافته، شما به مقاله‌های «برنامه‌نویسی ساخت‌یافته» دایکسترا [۳۳] و «برنامه‌نویسی ساخت‌یافته در سیستم‌های بزرگ» میلز [۱۹۶] ارجاع داده‌اید. در اینجا نیز دو قاعده را تعریف کرده‌اید:

«قاعده نخست می‌گوید که برنامه‌های ساخت‌یافته، از بالا به پایین و در سطوح مختلف تولید می‌شوند. سطوح یک برنامه ساخت‌یافته معمولاً همان سطوح انتزاع نیستند، زیرا از قاعده مالکیت منابع پیروی نمی‌کنند. قاعده دوم می‌گوید که کدام ساخت‌های کنترلی در برنامه‌های ساخت‌یافته ممکن است مورد استفاده قرار گیرند. تنها ساخت‌های کنترلی زیر مجاز می‌باشند: الحاق^{۴۵}، انتخاب دستور بعد بر پایه آزمون یک شرط، و تکرار^{۴۶}.» [۱۷۱]

مشتاقم که بدانم شما چگونه به رابطه بین سطوح انتزاع و برنامه‌نویسی ساخت‌یافته نگاه می‌کردید و چگونه دیدگاهتان را به مرور زمان پالایش کردید. اگر درست فهمیده باشم، در سیستم



شکل ۱- هر سطح شامل یک پودمان است. پیکان‌ها وابستگی بین پودمان‌ها را نشان می‌دهند

انتخاب کنید و چنین روش‌هایی به وجود آمده است. بنابراین، باید به این حرف‌ها در بافت زمانی خودش فکر کنید. در آن زمان، به خوبی نمی‌دانستند این کارها را چگونه انجام دهند و بسیاری از این حرف‌ها با توجه به آنچه امروز می‌دانیم، منطقی نیستند.

روشگان طراحی

دی‌لایت: مایلم درباره‌ی مقاله ۱۹۷۲ شما درباره‌ی روشگان طراحی با عنوان «روشگان طراحی برای سیستم‌های نرم‌افزاری اطمینان‌پذیر» [۱۷۱]، مطالبی را با شما در میان بگذارم. شما در این مقاله نوشته‌اید:

«مسائل به نحو مؤثری در خلال فرایند طراحی قابل حل هستند.» بر خلاف کارهایی که بعد از طراحی، مثلاً با روش‌های اشکال‌زدایی^{۴۱} می‌شد. کار دایکسترا بر روی سیستم THE نیز همراه با اثبات صحت بود. سطوح انتزاع به او کمک کرد تا صحت سیستمش را اثبات کند. آیا هنگامی که بر روی سیستم ونوس و سطوح انتزاعش کار می‌کردید، کار دایکسترا هیچ نقشی بر روی کار شما داشت و آیا شما با دایکسترا درباره‌ی این موضوع صحبت کرده بودید؟ شما همچنین در اصل، طرفدار «اثبات‌های غیرصوری^{۴۲} صحت برنامه» بودید. چرا بر روی صفت «غیرصوری» تأکید داشتید و نظر امروزتان چیست؟ **لیسکوف:** من هیچگاه با دایکسترا در این باره صحبت نکردم. با وجود این، احتمالاً گروهی که بر روی اثبات صوری برنامه‌ها کار می‌کردند بر روی من تأثیر گذاشته بودند، حتی اگر اعتقاد نداشتم که آن روش واقعاً عملی باشد. در مقاله‌ام راجع به روشگان طراحی، تلاش کرده‌ام نرم‌افزار را طوری سازمان‌دهی کنم که خواننده استدلال بهتری برای صحت آن به دست آورد، هر چند که این استدلال غیرصوری بود. من به کارایی استدلال صوری عقیده نداشتم. من طرفدار جدی استدلال غیرصوری هستم. مشکلی که با استدلال صوری وجود دارد این است که مشخصات^{۴۳} سیستم

44- theorem provers

45- concatenation

46- iteration

41- debugging

42- informal proofs

43- specification

ونوس، هر سطح انتزاع به یک پودمان^{۴۷} مربوط است، همان گونه که در شکل ۱ نشان داده شده است. این شکل بیانگر گراف وابستگی پودمان هاست: هر مستطیل یک پودمان را نشان می‌دهد، هر خط داخل پودمان، یک رویه را و هر پیکان از پودمان A به پودمان B، نشانگر وابستگی A به B از نظر کارکردی است.

لیسکوف: امروز که به آن مقاله فکر می‌کنم، به نظرم می‌رسد آنچه را که «سطوح انتزاع» نامیده بودم، مهم‌ترین چیز بوده است. من گاهی آن‌ها را «بخش»^{۴۸} نیز نامیده‌ام. هر بخش در واقع یک پودمان است. ایده اصلی این است که هر منبع دقیقاً متعلق به یک بخش از سیستم است و هیچکس دیگر نمی‌تواند به آن دسترسی داشته باشد. این چیزی است که به کارهای بعدی درباره نوع داده‌ها انتقال یافت و سرانجام به نوع داده انتزاعی^{۴۹} انجامید. من همچنین همواره طرفدار روش‌های برنامه‌نویسی بالا به پائین بوده‌ام که در پایان، گراف برنامه را در اختیار شما قرار می‌دهد و هیچ قاعده‌ای برای بازشناسی لایه اول، لایه دوم، لایه سوم و غیره وجود ندارد. امروز می‌گویم یک چرخه کوچک نیز اشکالی ندارد زیرا باید درباره آنچه با بازگشت^{۵۰} می‌کنید نیز فکر کنید. شما می‌توانید نمونه‌هایی از چنین گراف‌های وابستگی پودمان‌ها را در کتاب‌های من [۱۷۵، ۱۷۶]، به ویژه در کتاب دوم ببابید. شکل صفحه ۳۰۷ یک نمونه را نشان می‌دهد (شکل ۲ را ببینید). همان گونه که می‌بینید، یک چرخه وجود دارد و چیزی که من به دانشجوینام می‌گویم این است که چرخه‌ها نشانگر بازگشت هستند. اگر بازگشت متقابل^{۵۱} داشته باشید، این می‌تواند مسئله‌ساز باشد.

قاعده عبور نکردن از محدوده، چیزی است که در این مقاله اهمیت دارد. چیز دیگری که فکر می‌کنم نوآورانه و جدید بود، ایده داشتن چند رویه در هر پودمان بود. بنابراین هر پودمان تنها یک رویه نداشت که با آن بتوانید تعامل کنید، بلکه تعدادی رویه داشت. این مقاله درباره روش‌های طراحی است. بدین جهت تلاش کرده است که تمام جنبه‌ها را پوشش دهد. با این موضوع شروع کرده است که «باید طراحی را بالا به پایین انجام دهیم و ساخت‌های کنترلی خوب تعریف شده، کفایت خواهد کرد» اما در این مقطع من عقیده ندارم که شما آن گونه که ویرث پیشنهاد کرده، برنامه‌هایتان را بنویسید، یعنی برنامه‌های انتزاعی. در عوض، من فکر می‌کنم شما از طریق تعریف و نوآوری انتزاعی طراحی می‌کنید و کاری که می‌کنید این است که می‌گویید «اجازه بده به چگونگی پیاده‌سازی آن فکر کنم» - شما به سراغ نوشتن کد نمی‌روید. و شما ممکن است هر بار بیش از یک انتزاع معرفی کنید، زیرا من بر حسب یک ماشین انتزاعی فکر می‌کنم. البته همه این چیزها در آن مقاله نیست. آنچه اکنون به شما می‌گویم، گونه تکامل یافته آن روش‌ها

47- module

48- partition

49- abstract data type

50- recursion

51- mutual recursion

است که نقطه آغاز شکل‌گیری آن در همان مقاله بوده است.

دی لایت: آیا شما در همان زمان هم درباره برنامه‌نویسی ساخت‌یافته تردید داشتید؟ من به روشنی این سطوح انتزاع را در مقاله شما می‌بینم و فکر می‌کنم اگر آن بخش مربوط به برنامه‌نویسی ساخت‌یافته هم در آن نبود، مقاله هنوز به همان قوت و استحکام باقی می‌ماند.

لیسکوف: من فکر می‌کنم مهم‌ترین بخش مقاله، بخش مربوط به پودمان‌ها و این که چه شکلی هستند بود. بقیه‌اش در واقع تکمیل کننده همین داستان بود. اما من به تمام داستان علاقه‌مند بودم و فکر می‌کنم مقاله دو بخش داشت. این طور بود که هر چه پیش می‌رفتم، بیشتر بر روی پودمانگی^{۵۲} تمرکز می‌کردم زیرا به نظر می‌رسید جالب‌ترین مسئله برای کار کردن بود. احتمالاً تا اواخر دهه ۱۹۷۰ طول کشید تا دوباره برگشتم و کار بر روی بقیه مسئله، مثلاً این که «چگونه طراحی می‌کنید؟» و «چگونه مشخصات می‌نویسید؟»، را آغاز کردم. بنابراین، همه چیز از ابتدا آنجا بود، اما مسئله بزرگی بود، بزرگ‌تر از آن که بتوان یکباره به آن پرداخت. این مقاله سعی می‌کند که کل مسئله را نمایش دهد. این اتفاقی بود که برای پیشرفت و تکامل کار افتاد.

دی لایت: اگر بخواهم خلاصه کنم: در اوایل دهه ۱۹۷۰ شما طرفدار نرم‌افزار لایه‌ای، با چند رویه در هر لایه بودید، رویکردی بالا به پایین داشتید با احتمال دور زدن لایه‌ها. اما در یک مقطع، پاراناس سیستم‌های لایه‌ای و به ویژه سیستم THE را مورد بررسی دقیق قرار داد. او این کار را با صورت‌بندی «پایگان کاربری‌ها»^{۵۳} انجام داد [۲۳۳، ۲۳۴].

لیسکوف: خوب، پایگان کاربری‌ها و نمودارهای وابستگی پودمان‌های من خیلی شبیه به هم هستند. پایگان کاربری‌ها باعث این گراف در صفحه ۳۰۷ کتاب من شده است. (شکل ۲ را ببینید) **دی لایت:** که بسیار با یک سیستم نرم‌افزار لایه‌ای متفاوت است.

لیسکوف: کاملاً، و این نقطه‌ای است که من به آن رسیدم. من فکر می‌کنم اگر از دنیای سیستم عامل بیرون بیایید، نگاه نسبتاً متفاوتی به چگونگی کار چیزها پیدا خواهید کرد. سیستم عامل به طور کامل ماشین را پوشش می‌دهد و در نتیجه همانند بازنویسی کامل است. زبان برنامه‌نویسی نیز مثل همین است. شما زبان ماشین را دارید، زبان برنامه‌نویسی را دارید، بر روی زبان برنامه‌نویسی برنامه‌هایتان را اجرا می‌کنید و هرگز به ماشین زیرین دسترسی ندارید. از اینجا این ایده‌ها بیرون آمده است که باید پنهان‌سازی^{۵۴} کامل وجود داشته باشد. اما هنگامی که شروع به ساختن برنامه‌های کاربردی می‌کنید، دیگر شبیه این نیست. بنابراین، من فکر می‌کنم برخی از این ایده‌ها احتمالاً تحت تأثیر این واقعیت شکل گرفته‌اند که این افراد بر روی سیستم‌های عامل و زبان‌های برنامه‌نویسی کار

52- modularity

53- uses hierarchy

54- hiding

بنابراین، روشگانی که درباره آن در کتابم صحبت کرده‌ام این است که برنامه را می‌نویسید و سعی نمی‌کنید که بیش از حد آن را بهینه‌سازی کنید. فقط سعی می‌کنید که ساختار خوب و تمیزی داشته باشید و سپس از نظر کارایی آن را اشکال‌زدایی می‌کنید و تشخیص می‌دهید که موانع در کجا هستند. و آنگاه، با فرض یک ساختار پودمانی، به سادگی می‌توانید یک پودمان را بگیرید و آن را دوباره پیاده‌سازی کنید و تأثیری بر روی هیچ پودمان دیگری نخواهد داشت. بدین ترتیب، این در واقع شیوه خوبی برای به‌دست آوردن یک برنامه کارآمد است.

دی‌لایت: با تجربه‌ای که به‌عنوان پژوهشگر در سیستم‌های نهفته^{۵۹} کسب کرده‌ام، فکر می‌کنم که شکستن درون‌بندی بین پودمان‌ها می‌تواند به بهبود زیادی از نظر سرعت اجرا یا حافظه مصرفی بینجامد. در نتیجه، این می‌تواند یک روش نسبتاً حاد و غیرمعمول برای بهینه‌سازی کدهای کثیف به شمار آید. من نیاز به پودمانگی را درک می‌کنم اما چرا این دو مرحله را از هم جدا نکنیم؟ یعنی داشتن رویکردی پودمانی و به دنبال آن، مرحله‌ای که در آن ساختار پودمانی عمداً شکسته می‌شود.

لیسکوف: نخست آن که امروز برنامه‌های مترجم این کار را برای شما انجام می‌دهند. جایگذاری درون خطی^{۶۰} یک نمونه از آن است. (قراردادن کد تابع در نشانی هر فراخوانی تابع، توسط برنامه مترجم را جایگذاری درون خطی می‌نامند. مترجم) این ممکن است با آنچه شما می‌گویید تفاوت داشته باشد اما می‌تواند تنها از طریق ترجمه دو پودمان با هم و انجام بهینه‌سازی، باعث بهبود عمده‌ای از نظر کارایی گردد. به‌علاوه، برنامه‌های مترجم این روزها بهینه‌سازی‌های هوشمندانه زیادی انجام می‌دهند. اگر تصمیم بگیرید که یک گروه کوچک از پودمان‌ها را با یک پودمان یا گروه دیگری از پودمان‌ها جایگزین کنید نیز اشکالی ندارد. دقت کنید که اگر این کار را از طریق برنامه مترجم انجام دهید، کاملاً بی‌خطر است زیرا همیشه می‌توانید به ساختار پودمانی اولیه برگردید، در حالی که اگر این کار را دستی انجام دهید، از ساختار برنامه منحرف شده‌اید و احتمال دارد که بعداً با مشکل روبرو شوید. بنابراین بهتر است رویکرد محتاطانه‌تری داشته باشید.

موشکافی بیشتر در سیستم‌های لایه‌ای

دی‌لایت: در اواخر دهه ۱۹۶۰، سیلبرشاتز و گالوین [۲۷۹] اظهار کردند که حداقل دو مشکل در سیستم‌های لایه‌ای وجود دارد. نخست آن که چگونه لایه‌ها را تعریف و ترتیب‌دهی می‌کنید؟ ظاهراً اجماعی در این مورد وجود ندارد. اگر به کار دایکسترا در سال ۱۹۶۹ نگاه کنیم، او صریحاً اظهار کرده که ترتیب‌دهی او در سیستم THE لزوماً بهترین ترتیب‌دهی نبوده است. سیلبرشاتز و

می‌کنند، پس کاری کنیم که به شیوه‌های خاصی درباره چیزها فکر کنند.

ما در مورد پاراناس زیاد صحبت نکرده‌ایم اما او بسیار بر روی من تأثیرگذار بود و شاید بیشترین تأثیر را داشت. فکر می‌کنم او بود که این پرسش مهم را مطرح کرد که «پودمان‌ها چیستند؟». چیزی که در سخنرانی جایزه تورینگ گفتم این بود که روشگان برنامه‌نویسی در واقع درباره دو موضوع جداگانه است. یکی «چگونه طراحی می‌کنید؟» و دیگری «ساختار سیستم باید چه شکلی داشته باشد؟» **دی‌لایت:** چیزی که من از کارهای دایکسترا، ویرث و هور برداشت می‌کنم این است که در میانه‌های دهه ۱۹۶۰، فناوری الکترونیک در مقایسه با دهه ۱۹۵۰ پیشرفت بسیاری کرد و کارایی ماشین دیگر حائز بیشترین اهمیت نبود. در نتیجه، طراح نرم‌افزار مجاز بود که ابتدا به ساختار برنامه‌اش بیندیشد. به عبارت دیگر، شما مجاز بودید تلاش بیشتری را صرف به دست آوردن یک چیز ساخت‌یافته بکنید، حتی اگر حاصل کار از لحاظ کارایی ماشین در وضعیت بدتری قرار می‌گرفت.

لیسکوف: من فکر می‌کنم دیدگاه کمی متفاوت باشد. به اعتقاد من اتفاقی که افتاد این بود که بحران نرم‌افزار^{۵۵} به‌عنوان یک چیز فوق‌العاده مهم در نظر گرفته شد. به این جهت، خیلی مهم بود که شروع به استفاده از تکنیک‌ها و روش‌هایی کنند که احتمال کارکردن نرم‌افزار را بالا می‌بردند، و این لزوماً با تلاش برای حداکثر استفاده از کارایی ماشین، همساز نبود.

یکی از چیزهایی که برنامه‌نویسان معمولاً نگران آن بودند این بود که آیا برنامه‌های مترجم (کامپایلر) برای زبان‌های سطح بالاتر می‌توانند کد مقصد^{۵۶} کارآمدی تولید کنند یا نه. آن‌ها گاهی برنامه‌هایشان را دوباره به زبان ماشین می‌نوشتند تا پیاده‌سازی کارآمدتری به دست آورند. امروز برنامه‌های مترجم خیلی خوب کار می‌کنند. جایی که هنوز مشکل وجود دارد و برنامه‌های مترجم خیلی خوب عمل نمی‌کنند، زباله‌روبی^{۵۷} است. و البته در آن روزها این مسئله وجود نداشت زیرا برنامه‌نویسان به زبان الگول^{۵۸} و مشتقات آن برنامه می‌نوشتند و باید خودشان مدیریت حافظه را انجام می‌دادند. فناوری برنامه‌های مترجم پیشرفت کرده است اما بخشی از چیزهایی که آن موقع می‌گفتند این بود که «بله، شما مقداری از کارایی را از دست خواهید داد زیرا جملات goto ندارید یا از این زبان سطح بالاتر استفاده می‌کنید، اما از همه مهم‌تر این است که برنامه‌تان کار کند.» من فکر می‌کنم این اساسی‌ترین تغییر در درک افراد بود.

نکته دیگر این است که اگر برنامه‌ای را به‌صورت پودمانی بنویسید، آزادی عمل زیادی برای بازپیاده‌سازی^{۵۸} آن خواهید داشت.

55- Software Crisis

56- object code

57- garbage collection

58- re-implementation

59- embedded systems

60- in-line substitution

گالوین سی سال بعد نیز همان عقیده قبلی خود را ابراز داشته‌اند. نظر شما امروز در این باره چیست؟

لیسکوف: خوب، من به سیستم‌های لایه‌ای اعتقادی ندارم و احتمالاً یکی از دلایل بی‌اعتقادی من همین نکاتی است که شما از سیلبرشاتز و گالوین نقل کردید. حال توجه شما را به این نکته جلب می‌کنم که اگر لایه‌ها را کنار بگذارید و فقط بر پایه نمودار وابستگی پودمان‌ها که پیشتر به شما نشان دادم فکر کنید، همین اتفاق ممکن است دوباره بیفتد. به دست آوردن چنین نموداری به این معنی است که در جایی تصمیم طراحی خاصی گرفته‌ام که مرا به کشیدن این گراف به این شیوه خاص واداشته است. و اگر به نحو دیگری عمل کرده بودم، ممکن بود ساختار کاملاً متفاوت و شاید بهتری می‌داشتیم. پس شکی نیست که به هنگام طراحی، هرگاه که تصمیمی می‌گیرید، تعدادی از راه‌حل‌ها را از دور خارج می‌کنید. شما همچنان که در فرایند طراحی پیش می‌روید، همواره خود را محدودتر و محدودتر می‌کنید. این دقیقاً آن چیزی نیست که سیلبرشاتز و گالوین درباره‌اش حرف می‌زنند زیرا آن‌ها تلاش دارند آن چیزی را که من فکر می‌کنم یک ساختار مصنوعی است، اما تحمیل کنند و بعد درباره این ساختار مصنوعی بحث می‌کنند. اما حتی اگر این را در نظر نگیرید و به طراحی واقعی فکر کنید، هنوز هر بار که تصمیمی می‌گیرید، در واقع محدودیت‌های تازه‌ای را تعریف می‌کنید و امکاناتی را کنار می‌گذارید. این یکی از چیزهایی است که طراحی را سخت می‌کند.

دی‌لایت: آیا پژوهشی درباره صورت‌بندی گزینه‌های مختلف طراحی که پیش روی مهندس نرم‌افزار قرار دارد، انجام شده است؟

لیسکوف: خوب، آدم‌ها نوعاً طراحی‌های چندگانه انجام نمی‌دهند. شما معمولاً از پس بیش از یک طراحی بر نمی‌آیید. بنابراین، کاری که می‌کنید این است که سعی می‌کنید تصمیمات طراحی خود را توجیه کنید. در این راه، مثلاً می‌گویید «من درباره انجام این کار به این شیوه یا آن شیوه فکر کردم و این بهترین شیوه است زیرا ...». البته من از زمانی که در دهه ۱۹۸۰ کار بر روی روش‌گان برنامه‌نویسی را خاتمه دادم، بر روی سیستم‌های توزیعی^{۶۱} کار کرده‌ام و دیگر توجه چندانی به جامعه روش‌گان برنامه‌نویسی و جامعه مهندسی نرم‌افزار نداشته‌ام. بنابراین، ممکن است کارهایی صورت گرفته باشد اما من از آن آگاهی ندارم.

دی‌لایت: دومین اظهار نظر سیلبرشاتز و گالوین این بود که سیستم‌های لایه‌ای کارایی کمتری نسبت به انواع دیگر سیستم‌ها دارند. شاید بتوان یافته‌های آن‌ها را به نرم‌افزار پودمانی هم تعمیم داد و شاید هم نه. آن‌ها گفته‌اند:

«یک برنامه کاربر که دستور ورودی/خروجی اجرا می‌کند، در واقع یک فراخوانی سیستم^{۶۲} انجام می‌دهد که در لایه ورودی/خروجی قرار دارد و آن هم لایه مدیریت حافظه را از طریق لایه زمانبندی

پردازنده (CPU) و نهایتاً سخت‌افزار فراخوانی می‌کند. در هر لایه، پارامترها ممکن است اصلاح شوند، داده‌ها ممکن است نیاز باشد که منتقل شوند و نظایر آن. هر لایه، سرباری^{۶۳} را به فراخوانی سیستم می‌افزاید و نتیجه نهایی، فراخوانی سیستمی است که زمان بیشتری از هنگامی که سیستم لایه‌ای نبود، می‌گیرد. این محدودیت‌ها در سال‌های اخیر باعث واکنش منفی نسبت به لایه‌بندی شده‌اند. لایه‌های کمتری با کارکرد بیشتر، طراحی می‌شوند تا از بیشترین مزایای کدهای پودمانی استفاده شود و در عین حال از مشکلات تعریف و تعامل لایه‌ها اجتناب گردد.

به‌عنوان نمونه، تاریخچه ویندوز^{۶۴} ان‌تی را در نظر بگیرید. نخستین نسخه آن که عرضه شد، سازماندهی شدیداً لایه‌گرایی داشت. بدین خاطر، این نسخه، کارایی کمتری در مقایسه با ویندوز ۹۵ داشت. ویندوز ان‌تی ۴/۰ با انتقال لایه‌ها از فضای کاربر به فضای هسته سیستم عامل^{۶۵} و یکپارچه‌سازی آن‌ها، برخی از ناکارآمدی‌ها را جبران کرد.» [۲۷۹]

لیسکوف: فکر می‌کنم دوباره وارد سرزمین سیستم عامل شدیم. (خنده لیسکوف) برای من روشن نیست که مطلب گفته شده، عمومیت داشته باشد. اما شما می‌دانید که پودمانگی هزینه دارد. این در واقع هزینه فراخوانی روتیه‌ها نیست، زیرا برنامه مترجم به راحتی می‌تواند از شرش خلاص شود. هزینه زیاد می‌شود زیرا تصمیم طراحی چنین ایجاب کرده و در نتیجه، به جای آن که به دو چیز با هم فکر کنم، به‌طور جداگانه فکر می‌کنم. با وجود این، من هنوز به روش‌گانی که می‌گوید اشکال‌زدایی کارایی انجام بده و سپس بین چه چیز را باید تغییر دهی، اعتقاد دارم.

به‌علاوه، من فکر می‌کنم در حوزه دانش رایانه (علوم کامپیوتر)، درباره کارایی مبالغه شده است، زیرا چیزی که شما از کارایی می‌خواهید این است که به قدر کافی خوب باشد. شما به بهترین کارایی نیاز ندارید. نیازهای کارایی به دلیل کمبود زبان‌های برنامه‌نویسی خوب، جدی‌تر شده است. بنابراین، خیلی اهمیت دارد که زبان‌هایی که انتخاب می‌کنید، پیاده‌سازی خوبی داشته باشند. خیلی اهمیت دارد که سیستم عامل شما پیاده‌سازی خوبی داشته باشد. این که خدمات وب^{۶۵} شما پیاده‌سازی فوق‌العاده‌ای داشته باشد از اهمیت کمتری برخوردار است زیرا اکنون شما بر روی سرعت برنامه‌هایی کار می‌کنید که برای مردم ملموس باشد و این به شما انعطاف‌پذیری زیادی می‌دهد.

مشخصات: صوری در مقابل غیر صوری

دی‌لایت: در مورد مشخصات صوری، همکار من مارک پریستلی چنین نوشته است:

«لیسکوف و زیلس همچنین ارتباطی قوی بین استفاده از نوع

63- overhead

64- kernel

65- web services

61- distributed systems

62- system call

و می‌دیدند که این‌ها صرفاً ایده‌های انتزاعی نیستند و به منظور اشکال‌زدایی نیز می‌توانند مورد استفاده قرار گیرند.

دی‌لایت: برتراند مهیر با زبان ایفل (یکی از زبان‌های برنامه‌نویسی شیء‌گرا، مترجم) نیز کم و بیش به همین شیوه کار می‌کند. من فکر می‌کنم کار او بیشتر صوری است.

لیسکوف: او به شما اجازه می‌دهد که چیزهایی را که جملات توضیحی^{۶۹} نیستند اما توسط برنامه مترجم تفسیر شده‌اند در متن برنامه بگنجانید. پس البته که باید صوری باشند.

دی‌لایت: گروه کاری ۲/۳ فدراسیون بین‌المللی پردازش اطلاعات (IFIP) خیلی درگیر روش‌گان برنامه‌نویسی صوری بود، مگر نه؟

لیسکوف: بله و برای مدت کوتاهی من عضو آن گروه کاری بودم. شاید هم عضو نبودم اما آن‌ها به من به چشم یک عضو بالقوه نگاه می‌کردند. درست یادم نمی‌آید. من شخصاً تمام آن گروه‌های کاری را چندان مفید نمی‌دانستم.

زبان CLU

دی‌لایت: زبان CLU (این زبان برنامه‌نویسی که در سال‌های ۱۹۷۴ و ۱۹۷۵ در ام‌آی‌تی توسط باربارا لیسکوف و دانشجویانش درست شد، یکی از زبان‌های پیش‌تاز است که بسیاری از ویژگی‌هایی که امروز به‌طور گسترده‌ای مورد استفاده هستند برای نخستین بار در آن زبان معرفی شده‌اند. زبان CLU به‌عنوان گام مهمی در توسعه برنامه‌نویسی شیء‌گرا در نظر گرفته می‌شود. نام این زبان از CLUs-ter به معنی خوشه گرفته شده است. مترجم) که شما آن را به وجود آوردید، دارای انتزاع داده‌ها، انتزاع تکرار، مدیریت استثنائات^{۷۰} و نوع‌های چندریختی^{۷۱} بود. مایلم بدانم چگونه گروه پژوهشی شما در خلال دهه ۱۹۷۰ به این ایده‌ها دست یافت. شمادر خاطرات خود در سال ۱۹۹۳ [۱۴۷] به سیمولا ۶۷ و اسمال‌تاک^{۷۲} در این رابطه اشاره کرده‌اید.

لیسکوف: من خیلی به رویکرد گروه اسمال‌تاک توجهی نداشتم زیرا عمدتاً دربارهٔ وراثت^{۷۳} بود. کارهای اولیه آن‌ها خیلی بر پیاده‌سازی متمرکز بود و انتزاعی وجود نداشت. تماماً دربارهٔ رده‌ها و زیررده‌ها^{۷۴} بود: ساختن یک زیررده از یک رده، افزودن چند حوزه^{۷۵} و چند روش^{۷۶} اضافی، تغییر رفتار. حفظ رفتار اُبرده^{۷۷} برای آن‌ها اهمیت نداشت. همچنین، من به سیمولا ۶۷ نیز علاقه‌مندی خاصی نداشتم زیرا سیمولا ۶۷ در آن زمان فاقد درون‌بندی بود.

داده‌های انتزاعی و جنبه‌های مربوط به اثبات صحت برنامه‌ها دیده‌اند. [...] آن‌ها شیوه‌های مختلف ارائهٔ مشخصات صوری انتزاع داده‌ها را در نظر گرفته‌اند.» [۲۴۶]

لیسکوف: استیو زیلس برای پایان‌نامهٔ دکتریش بر روی مشخصات جبری برای نوع داده‌های انتزاعی کار می‌کرد. و موضوع پایان‌نامهٔ دکتریش جان گوتاگ، که با من در نوشتن کتابم همکاری کرد، نیز همین بود. من هیچگاه خودم بر روی آن کار نکرده‌ام. فصل دهم کتابم با عنوان «نوشتن مشخصات صوری» [۱۷۵]، عمدتاً برآمده از زبان Larch گوتاگ [۱۱۵] است.

دی‌لایت: کتاب اخیر شما [۱۷۶] این حس را در من به‌وجود آورده است که شما دربارهٔ مشخصات صوری تردید دارید.

لیسکوف: اوه، من از قبل هم تردید داشتم. من فکر می‌کردم این کار جالبی است و یک دانشجوی دکتریش به نام والدیس برزینس داشتم که بر روی یک مدل انتزاعی برای ارائهٔ مشخصات صوری کار می‌کرد و من اعتقاد داشتم که نویدبخش‌تر از شیوهٔ جبری است. من حس می‌کردم که شیوهٔ جبری آنقدر انتزاعی است که مردم برای به کارگیری آن با مشکلات زیادی روبرو خواهند شد. مشکلی که با هر شیوهٔ صوری وجود دارد این است که نوشتن مشخصات صوری برای افراد بسیار دشوار است.

دی‌لایت: در سال ۱۹۷۵، شما و زیلس نوشته‌اید: «مشخصات صوری به‌عنوان یک رسانهٔ ارتباطی برتر از مشخصات غیرصوری است.» [۱۷۷]

لیسکوف: (خندهٔ لیسکوف) مشکل اینجاست که مشخصات غیرصوری می‌تواند تعبیرهای گوناگونی داشته باشد و دانستن این که واقعاً چه می‌گوید، دشوار است. اما مشخصات صوری هم می‌تواند کاملاً غیرقابل درک باشد. یعنی شما به آن نگاه کنید و مطلقاً نفهمید چه می‌گوید.

دی‌لایت: بدین ترتیب، آیا این امروز یکی از مشکلات عمدهٔ نرم‌افزار نیست؟

لیسکوف: بله، همین طور است. نه تنها مشخصات صوری، بلکه حتی مشخصات غیرصوری هم مشکل‌زا هستند. اگر شما برگهٔ (فرم) خاصی برای بیان آن‌ها داشته باشید، کمک کننده است. در کتاب من [۱۷۶]، برگه‌ای برای بیان مشخصات غیرصوری وجود دارد. به دانشجویان آموزش می‌دادم که مشخصات را در آن برگه‌ها بنویسند. افزون بر این، دانشجویان باید تابع انتزاعی و مقدار ثابت بازنمود^{۶۶} را پیاده‌سازی می‌کردند. (تابع انتزاعی، تابعی است از بازنمود غیرانتزاعی و عینی شیء^{۶۷} به مقدار انتزاعی که نشان می‌دهد. مقدار ثابت بازنمود، شرطی است که باید برای تمام بازنمودهای عینی و معتبر یک رده^{۶۸}، برقرار باشد. مقدار ثابت بازنمود قلمرو تابع انتزاعی را نیز تعریف می‌کند. مترجم) این کار برای آن‌ها بسیار مفید بود

69- comment

70- exception handling

71- polymorphic types

72- Smalltalk

73- inheritance

74- subclasses

75- field

76- method

77- super class

66- representation invariant

67- object

68- class

تمرکز اولیه آن‌ها بر شبیه‌سازی^{۷۸}، همروندی^{۷۹} و نظایر آن بود. البته آن‌ها سازوکار وراثت را داشتند که بعد توسط جامعه شیء‌گرایی وام گرفته شد. گروه اسمال‌تاک بر روی مسئله متفاوتی کار می‌کردند. من بر روی انتزاع کار می‌کردم، آن‌ها بر روی یک روش پیاده‌سازی که به شما اجازه می‌داد روش‌های چندگانه داشته باشید. بنابراین می‌توانید مشاهده کنید که این دو ایده به همدیگر مربوطند اما تمرکزشان متفاوت است.

از دید من، CLU سه شکل انتزاع داشت: انتزاع رویه‌ای، انتزاع داده‌ها و انتزاع تکرار. چند ریختی، روش تعمیمی برای هر یک از اشکال انتزاع است. من مجبور نیستم مجموعه اعداد صحیح را تعریف کنم، می‌توانم مجموعه‌ای از نوع T داشته باشم. به همین ترتیب، می‌توانم تکرارگری^{۸۰} داشته باشم که توسط یک نوع T برحسب پارامتر یا پارامترها توصیف شده باشد. استثنائات نیز نوعی از تعمیم هستند. تنها به رویه‌ها و تکرارگرها مربوط می‌شود و راهی است برای این که مطمئن شوید که آن‌ها را با ضعیف‌ترین پیش‌شرط^{۸۱} قابل کنترل تعریف کرده‌اید و در عین حال، انواع مختلف نتایجی که ممکن است پیش آید را تشخیص داده‌اید. بنابراین، استثنائات از نظر من اهمیت داشتند زیرا به‌طور ایده‌آل می‌خواستیم که به‌عنوان پیش‌شرط، مقدار «درست»^{۸۲} را داشته باشیم اما می‌خواستیم از راه‌حل C که شما صرفاً همه چیز را در جواب می‌گنجانید، اجتناب کنیم. البته مشکلی که با آن وجود دارد این است که خیلی مستعد خطاست زیرا مردم حتی به خودشان زحمت نگاه کردن به جواب را نمی‌دهند. من استثنائات را به منظور توجه‌دادن به فراخوانده که چیزی غیرعادی اتفاق افتاده است، می‌خواستیم. هنگامی که پروژه CLU را آغاز کردم، تکرارگرها در برنامه کار نمی‌کردند. آن‌ها بعداً هنگامی که دریافتیم به چیزی برای تکمیل داستان انتزاع داده‌ها نیاز داریم که مستلزم ایجاد انتزاع‌های بد یا نوشتن برنامه‌های بد توسط برنامه‌نویسان نباشد، وارد صحنه شدند.

دی‌لایت: منظورتان از بد چیست؟

لیسکوف: من حالا از دید یک طراح زبان فکر می‌کنم. من می‌دانستم که قابلیت تکرار بر روی اشیاء مهم خواهد شد زیرا بسیاری از اشیاء، مجموعه‌هایی از چند چیز هستند و شما می‌خواهید که کاری را با تمام عناصر مجموعه انجام دهید. در غیاب تکرارگرها، من دو راه‌حل متفاوت در ذهن داشتم. یکی این بود که مجموعه را خالی کنید و بدین ترتیب بفهمید که کار تمام شده است. مشکلی که با این راه‌حل وجود داشت این بود که گاهی اوقات شما می‌خواستید دوباره از آن استفاده کنید که این به برنامه زشتی می‌انجامید و

یا می‌توانستید با تبدیل مجموعه به یک مجموعه شاخص‌دار^{۸۳} به نحوی که بتوانید به راحتی بر روی آن عملیات تکراری داشته باشید، انتزاع را پیچیده‌تر کنید. اما از دید زبان، من نمی‌خواستم مردم انتزاع‌های پیچیده‌تری از آنچه نیاز داشتند انجام دهند. بدین خاطر، در آن زمان نگران این بودم که چگونه راهی برای طراحان باز کنم تا ساده‌ترین انتزاع‌هایی را که قدرت بیانگری کافی داشته باشد، ببینند. انتزاع داده‌ها در حد کمال باید به قدر کافی عملیات در اختیار بگذارد که هر چیزی که کاربر نیاز به انجامش داشته باشد، به آسانی و با کارآمدی انجام‌پذیر باشد.

تکرارگرها از دل بحث‌هایی که با گروه آلفارد در کارنگی ملون [۳۱۳] داشتیم بیرون آمدند. آن‌ها بر روی مولدها^{۸۴} کار می‌کردند که بی‌شباهت به مولدهایی که امروز در جاوا داریم نبودند. با وجودی که مولدها مسئله را حل می‌کردند، ما فکر می‌کردیم که قشنگ نیستند. یکی از دانشجویان من به نام راس اتکینسون، در هواپیمای برگشت از کارنگی ملون و پس از آن که با ایده مولدها آشنا شده بودیم، تکرارگرها را ابداع کرد. تکرارگرها قدرت بیانگری کمتری نسبت به مولدها داشتند زیرا انواع خاصی از تکرارها وجود دارند که قابل بیان توسط آن‌ها نیستند، مانند مقایسه دو درخت برای بررسی این که عناصر آن‌ها برابرند یا نه. ما به دنبال سازوکارهای پیچیده‌تری بودیم اما در انتها به یک چیز ساده رسیدیم و از این قاعده مهندسی استفاده کردیم که بهتر است ۹۵٪ مسائل با یک سازوکار ساده پوشش داده شود تا آن که ۱۰۰٪ مسائل با یک سازوکار پیچیده‌تر حل گردد.

دی‌لایت: پس گروه شما یک رویکرد واقع‌گرایانه و عملی را دنبال کرد. با وجود این، زیبایی و ظرافت نیز به مقدار زیادی در نظر گرفته شده بود.

لیسکوف: دقیقاً. تکرارگرها یک راه‌حل بسیار زیبا بودند. ما تشخیص دادیم که به قدرت مولدها نیستند. بعد به فکر تکرارگرهای پیچیده‌تری افتادیم که قدرت بیشتری داشته باشند. و در پایان، به دنبال سازوکاری رفتیم که هم زیبا بود و هم به قدر کافی بیانگر.

این به‌طور خلاصه، هنر طراحی است. در گروه من، همیشه سعی می‌کردیم تمام جنبه‌های یک تصمیم را در نظر بگیریم به نحوی که در انتها بتوانیم تصمیم آگاهانه‌ای بگیریم. ما به‌صورت گروهی کار می‌کردیم و غالباً به اتفاق نظر می‌رسیدیم. البته گاهی اوقات به دو راه‌حل متفاوت می‌رسیدیم و یک نفر باید تصمیم نهایی را می‌گرفت. آن نفر، من بودم. و بله، زیبایی خیلی اهمیت داشت. **دی‌لایت:** نیرو یا نیروهای مقابل زیبایی چه بودند؟ برای مثال، آیا مهلت زمانی^{۸۵} برای اتمام پروژه، چنین نقشی نداشت؟

لیسکوف: این یک کار پژوهشی بود. بنابراین مهلت زمانی برای آن وجود نداشت. ما البته بر روی پیشرفت کار، خیلی تمرکز

83- indexed
84- generators
85- deadline

78- simulation
79- concurrency
80- iterator
81- precondition
82- true

راس اتکینسون باید دکتریش را بر روی تکرارگرها گرفته باشد. یکی از آن‌ها باید مدرک دکتریش را بر روی نوع‌های پارامتری شده گرفته باشد. اما من به انجام این کار به آن شیوه فکر نمی‌کردم. مطمئناً چند موضوع دکترای در این پروژه وجود داشت. اما ما همگی با هم بر روی زبان کار می‌کردیم و من کارها را به آن شکل، تفکیک نمی‌کردم.

دی‌لایت: شاید علت موفقیتش هم همین بود.

لیسکوف: شاید. اما من فکر می‌کنم با وجودی که یک پروژه گروهی بود، می‌توانستم آن را به بخش‌های مختلفی بشکنم. اما در آن زمان درک نمی‌کردم که می‌توان این کار را انجام داد.

دی‌لایت: آیا گروه‌های دیگر هم به همین شیوه کار می‌کردند؟ و امروز چطور؟ آیا هنوز به همین شیوه کار می‌کنید؟

لیسکوف: من درباره‌ی گروه‌های دیگر اطلاعی ندارم. شیوه‌ی کار گروه CLU تا دهه‌ی ۱۹۹۰ ادامه یافت. من جلسات هفتگی با گروه داشتم، افراد می‌آمدند و درباره‌ی مسائل اصلی صحبت می‌کردیم. امروز دیگر بدان شیوه کار نمی‌کنم زیرا گروه فعلی خیلی کوچک‌تر شده است. **دی‌لایت:** در رابطه با زیبایی یا سادگی، فرض من بر این است که هنگامی که دانشجوی جدیدی به گروه شما می‌پیوندد، ابتدا باید برایش جا بیفتد که این‌ها به چه معنی هستند. درست است؟ من به سختی می‌توانم تصوّر کنم که همه برداشت یکسانی از این واژه‌ها داشته باشند.

لیسکوف: من آموزش خاصی به آن‌ها نمی‌دهم. گروه امروزی من با آنچه در دهه‌ی ۱۹۹۰ بود، تفاوت دارد. قبلاً گروه بزرگی بود و دانشجویان جوان وارد آن می‌شدند و با دانشجویان قدیمی‌تر کار می‌کردند. دانشجویان قدیمی همگی دستیاران آموزشی من در کلاس بودند و بنابراین همه آن‌ها کاملاً با این روش‌گان آشنا بودند. نوعی یادگیری حسی بود. امروز، من با دانشجویانم کار می‌کنم و همیشه به آن‌ها می‌گویم «شاید این خیلی پیچیده باشد، بهتر است به راه ساده‌تری فکر کنیم»، زیرا زیبایی و سادگی خیلی به هم ارتباط دارند. هنگامی که بتوانید به راه‌حل ساده‌تری دست یابید، دلتان نمی‌خواهد که راه‌حل پیچیده‌ای را اختیار کنید.

دی‌لایت: آیا درباره‌ی سادگی با دیگر پژوهشگران هم بحث می‌کنید؟ معیار سادگی برای دایکسترا، در برهه‌ای از فعالیت حرفه‌ایش، داشتن اثبات صحت کوتاهی برای برنامه بود. و معیارهای دیگری هم برای سادگی وجود دارد. شما هنگامی که مثلاً به گروه کاری IFIP رفتید، چنین چیزی را حس کردید؟

لیسکوف: به یاد نمی‌آورم. چیزی که حس کردم این بود که آن‌ها به موضوعاتی که مورد علاقه من است، علاقه‌مند نیستند. شاید آن‌ها بیشتر در پی روش‌های صوری بودند. اما از حدود سال ۱۹۸۵، من دیگر به هیچ همایشی نرفته‌ام بجز آن‌هایی که مربوط به سیستم‌ها بوده‌اند. در اینجا، ما زیاد درباره‌ی این چیزها صحبت نمی‌کنیم. اما هنگامی که با اعضای گروه‌م درباره‌ی مقالات پژوهشی

داشتیم. زیبایی و قدرت بیانگری، نیروهای مقابل هم بودند. منظور از قدرت بیانگری، ساده‌سازی نوشتن برنامه‌ها بود. ما نشانه‌هایی برای برنامه‌های خوب در مقابل برنامه‌های بد داشتیم. این معیار ما بود. ما نمی‌خواستیم نوشتن هر برنامه‌ای را آسان‌تر کنیم. برای نمونه، اگر شما می‌خواستید از جملات goto در برنامه‌هایتان استفاده کنید، نمی‌توانستید این کار را در CLU به راحتی انجام دهید با وجودی که مطمئنم می‌توانستید ساختارهایی را برای انجام این کار ابداع کنید. بنابراین، ما نشانه‌ای برای یک رده از برنامه‌ها که می‌خواستیم از آن‌ها پشتیبانی کنیم، داشتیم. گرچه، احتمالاً نمی‌توانم به شما بگویم که آن رده چه بود. بعد مسئله عملکرد و کارایی مطرح می‌شد. ضمناً بگویم که تکرارگرها پیاده‌سازی بسیار کارآمدی داشتند و کار با آن‌ها نیز ساده بود. این‌ها جزء اهداف طراحی ما بودند و همواره مورد نظر قرار می‌گرفتند. ما همیشه برنامه‌های کوچکی می‌نوشتیم و سعی می‌کردیم بررسی کنیم که بیان یک چیز چقدر آسان یا سخت است.

دی‌لایت: آیا از آغاز از این چهار معیار آگاهی داشتید؟ یا به تدریج این معیارها به دست آمدند؟

لیسکوف: من فکر می‌کنم زیبایی آنقدر اساسی و بنیادی بود که آن را به عنوان یک اصل طراحی به حساب نمی‌آوردیم، اما بقیه، به عقیده من، از پیش تعیین شده بودند. سادگی، معیار دیگری است که با زیبایی تفاوت دارد. منظور از سادگی این است که مردم بتوانند درک کنند که شما درباره‌ی چه چیزی صحبت می‌کنید. سهولت استفاده این است که: آیا به راحتی می‌توانم برنامه‌هایی را با استفاده از این سازوکار بنویسم؟ قدرت بیانگری این است که: آیا می‌توانم تمام چیزهایی را که نیاز دارم بیان کنم. به راحتی بیان کنم؟ در ارتباط با کارایی، CLU یک زبان با زبانه‌روبی بود. بنابراین، ما می‌دانستیم که عملکرد آن همانند C نخواهد بود. (در زبان C، تخصیص و آزادسازی حافظه به صورت دستی توسط برنامه‌نویس صورت می‌گیرد و در نتیجه، از نظر زمانی، سرباری برای زبانه‌روبی خودکار توسط سیستم ندارد. مترجم) اما هنوز خیلی مواظب کارایی آن بودیم.

دی‌لایت: آیا تک‌تک دانشجویان شما همه این معیارها را در ذهن داشتند؟ این‌طور نبود که یکی از دانشجویان بر روی سادگی کار کند و دیگری بر روی کارایی؟

لیسکوف: نه، قطعاً نه. ما بر روی ساخت‌های زبان کار می‌کردیم و درباره‌ی برنامه‌های نمونه فکر می‌کردیم و این که چگونه آن‌ها را به این شیوه و یا آن شیوه بیان کنیم. در پایان، درباره‌ی این که راه‌حل چه باشد تصمیم می‌گرفتیم و با وجودی که به‌طور مشخص در مورد این هدف‌ها صحبت نمی‌کردیم اما آن‌ها را در ذهن داشتیم. **دی‌لایت:** شما وظایف مختلف را چگونه به دانشجویان محول می‌کردید؟ همه باید مدرک دکترای می‌گرفتند، درست است؟

لیسکوف: من در مورد تخصیص وظایف خوب عمل نمی‌کردم.

بحث می‌کنیم، غالباً صحبت‌هایی از قبیل این که رابط^{۸۶} سیستم قشنگ نیست یا پیاده‌سازی دارای این اشتباه است، مطرح می‌شود. بله، این صحبت‌ها در بحث‌های ما به میان می‌آید.

ایدا و محیط دانشگاهی امروز

دی‌لایت: آیا درست است که شما با یکی از تیم‌های زبان ایدا همکاری داشتید؟

لیسکوف: بله، من به یکی از تیم‌های صنعتی در ناحیه بوستن مشاوره می‌دادم. در اصل، چهار تیم بودند و همگی باید از بخش صنعت می‌بودند. هیچکدام دانشگاهی نبودند.

دی‌لایت: از نظر فنی، برداشت شما درباره ایدا چه بود؟

لیسکوف: ایدا خیلی پیچیده بود و در زیر بار سنگین مشخصاتش خم شده بود. در واقع، دو مسئله وجود داشت. طراحان انعطاف‌پذیری چندانی نداشتند زیرا شرط‌ها و محدودیت‌های وحشتناکی بر روی راه‌حل‌های قابل پذیرش گذاشته شده بود. در واقع، به نوعی می‌توان گفت که طراحی زبان توسط افرادی در وزارت دفاع که احتمالاً ایده درستی از آنچه انجام می‌دادند نداشتند، صورت گرفته بود. مسئله دوم این بود که بهترین طراحان زبان در طراحی این زبان دخالت نداشتند زیرا نیازسنجی توسط تیم‌هایی از بخش صنعت صورت گرفته بود. بنابراین، شما باید با مشخصاتی که نیازهای بسیاری داشت شروع می‌کردید و امکان بسیار کمی وجود داشت که بگوئید «خوب، ما نباید این کار را بکنیم، باید آن کار را بکنیم.» در واقع، من حس می‌کردم آن طرحی که برنده شد، بسیار فرمانبردار بود!

دی‌لایت: در رابطه با محیط دانشگاهی امروزی، مایلیم مطلبی را از ویرث نقل کنیم:

«در این محیط دانشگاهی پسامدرن، استاد مدتهاست که دیگر آن آدم دانشمند و خردمندی که به مطالعات بی‌سروصدای خود می‌پرداخت و عمیق‌تر و عمیق‌تر در موضوع مورد علاقه خود به غور و تحقیق می‌پرداخت، نیست. استاد امروزی، مدیر یک تیم بزرگ پژوهشی است که برای تأمین بودجه تیمش، رابطه نزدیکی با بنگاه‌های اصلی سرمایه‌گذاری برقرار می‌کند و به‌طور خستگی‌ناپذیری به نوشتن پیشنهاد پروژه‌های جالب و داستان‌های موفقیت عجیب و غریب می‌پردازد.» [۳۱۰]

لیسکوف: من فکر می‌کنم او درباره دانشگاه خودش (ETH) صحبت کرده است. در آنجا گروه‌های بزرگ با تعداد زیادی پژوهشگر دوره پسادکتری و یک استاد در رأس آن، وجود دارد. نه تنها بودجه خوبی از ETH دریافت می‌کنند بلکه باید به دنبال منابع مالی خارجی نیز بروند. صحبت او درباره دانش رایانه (علوم کامپیوتر) به طور کلی نیست.

دی‌لایت: شما وضعیتی را که در ام‌آی‌تی هست، چگونه توصیف می‌کنید؟

لیسکوف: در اینجا اغلب گروه‌ها نسبتاً کوچک هستند و با استاندارد آن‌ها (ETH) خیلی کوچک. بزرگ‌ترین گروه من ۹ دانشجو داشت. هر از گاهی من یک پژوهشگر دوره پسادکتری داشتم. مدل کاری من این بوده است که همواره از نزدیک به‌طور مستقیم با دانشجویان کار کرده‌ام، نه از طریق واسطه‌ها. بسیاری از استادان در اینجا گروه‌های کوچکی دارند، شش دانشجو با یک یا دو پژوهشگر پسادکتری و به‌صورت یک تیم کار می‌کنند. استاد عمیقاً درگیر فعالیت‌های علمی و پژوهشی است و مقدار پولی که آورده می‌شود برای پشتیبانی از تیم کافیت. نه بیشتر و بیشتر و بیشتر.

دی‌لایت: پس مشکلی برای تأمین بودجه پژوهشی وجود ندارد؟

لیسکوف: تأمین بودجه، دشوارتر از آن چیزی است که باید باشد. ما عادت کرده‌ایم که بیشتر بودجه مورد نیازمان را از دارپا (بنگاه پروژه‌های پیشرفته پژوهشی، وابسته به وزارت دفاع آمریکا. مترجم) بگیریم که در دهه ۱۹۹۰ مسئله‌ساز شده بود اما اکنون وضعیت عادی شده است.

دی‌لایت: راستش را بگوئید، آیا تأمین بودجه در ام‌آی‌تی راحت‌تر از سایر دانشگاه‌ها در آمریکا نیست؟

لیسکوف: شاید. اما آدم‌ها در اینجا وقت بیشتری را نسبت به قبل صرف نوشتن پیشنهاد می‌کنند. اکنون وضع کمی بهتر از سه یا چهار سال پیش شده است. بنیاد ملی علوم (NSF) تنها به ۱۰٪ پیشنهادهای وارده بودجه تخصیص می‌داد و دارپا تقریباً به‌طور کامل بر روی کارهای صنعتی متمرکز شده بود. بنابراین، در آن سال‌ها، مشکلات جدی وجود داشت.

دی‌لایت: ویرث همچنین می‌گوید که امروز از نظر فنی وضعیت «نومیدکننده‌ای» وجود دارد. او از C، C++، C# و جاوا به‌عنوان زبان‌های ضعیفی نام می‌برد که دانشگاهیان مجبورند از آن‌ها استفاده کنند.

لیسکوف: بله، اما چیزی عوض نشده است. در واقع، اکنون انتخاب و به‌کارگیری یک زبان پژوهشی کمی آسان‌تر است زیرا می‌توانید نرم‌افزار را به‌صورت متن‌باز^{۸۷} بر روی اینترنت بگذارید. پاسکال، بیش از هر زبان دانشگاهی دیگر، اجرای واقعاً خوبی داشت. اما به‌طور کلی، زبان‌هایی تا کنون به‌طور گسترده به کار گرفته شده‌اند که مورد پشتیبانی بخش صنعت قرار داشته‌اند. امروز زبان‌های تازه‌ای نیز مانند پایتون، PHP و روبی- که ممکن است زبان‌های خیلی خوبی هم نباشند- وارد صحنه شده‌اند. آن‌ها نه از محیط‌های دانشگاهی بیرون آمده‌اند و نه از روند کلی و خط فکری غالب در بخش صنعت.

دی‌لایت: باربارا لیسکوف، به خاطر وقتی که برای این گفتگو اختصاص دادید از شما تشکر می‌کنم.

لیسکوف: خواهش می‌کنم.

(ادامه دارد)

سنجش شباهت معنایی واژگان در زبان فارسی*

نسرین تقی‌زاده

آزمایشگاه پردازش هوشمند متن و زبان طبیعی، دانشکده مهندسی برق و کامپیوتر، دانشکده فنی دانشگاه تهران
پست الکترونیکی: nsr.taghizadeh@ut.ac.ir

نفیسه یاوری

آزمایشگاه پردازش هوشمند متن و زبان طبیعی، دانشکده مهندسی برق و کامپیوتر، دانشکده فنی دانشگاه تهران
پست الکترونیکی: n.yavari@ut.ac.ir

هشام فیلی

استادیار دانشکده مهندسی برق و کامپیوتر، دانشکده فنی دانشگاه تهران
پست الکترونیکی: hfaili@ut.ac.ir

چکیده

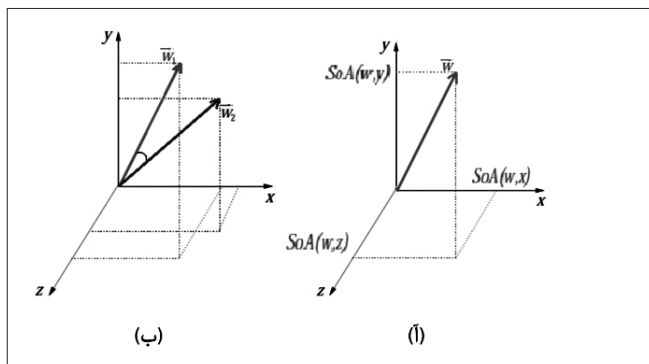
سنجش شباهت معنایی واژگان از جمله مسئله‌های مهم در پردازش متن و زبان طبیعی است که در بسیاری مسائل نظیر ساخت وردنت و استخراج رابطه از متن کاربرد دارد و معیارهای زیادی برای این منظور پیشنهاد شده است. اما تاکنون مقایسه جامعی بین آن‌ها در زبان فارسی صورت نگرفته است. در این مقاله دقت چندین معیار شباهت‌سنجی توزیعی مورد بررسی قرار گرفته است و با تحلیل نقاط قوت و ضعف آن‌ها، چندین معیار ترکیبی ارائه شده است. آزمایش‌ها نشان می‌دهند که PMI با ۴۲٪ بیشترین دقت را بین معیارهای توزیعی دارد و بهترین معیار ترکیبی ارائه شده توانسته است دقت آن را ۱۶٪ بهبود بخشد. کلمات کلیدی: شباهت‌سنجی معنایی، روش‌های توزیعی، اطلاعات متقابل نقطه‌ای، فاصله اقلیدسی، فاصله کسینوسی، آنتروپی، پیکره متنی.

۱- مقدمه

بین واژه‌ها و ابهام‌زدایی از واژه‌های متن به اندازه‌گیری هر چه دقیق‌تر شباهت معنایی واژه‌ها نیاز دارد. هم‌چنین در بازیابی اطلاعات، یکی از چالش‌های اصلی این است که پاسخ‌ها از لحاظ معنایی به پرسش کاربر مربوط باشند. لذا ارائه روش‌های کارآمد برای سنجش شباهت معنایی واژه‌ها در کاربردهای مختلف پردازش زبان‌های طبیعی مانند

سنجش شباهت معنایی واژه‌ها از مسئله‌های مهم در وب‌کاوی، بازیابی اطلاعات، و پردازش زبان‌های طبیعی است. تشخیص رابطه معنایی

* این مقاله در بیست و یکمین کنفرانس ملی سالانه انجمن کامپیوتر ایران، اسفند ۹۴ ارائه شده است



شکل ۱: (آ) واژه هدف در فضای هم‌رخدادی، (ب) فاصله بردارهای هدف [۸]

سازمان‌دهی این مقاله به صورت زیر است: بخش ۲ به معرفی معیارهای شباهت‌سنجی توزیعی می‌پردازد. بخش ۳ به ارزیابی و مقایسه دقت معیارهای مختلف اختصاص دارد. بخش ۴ مسئله ارائه معیار ترکیبی با دقت بهتر را مورد مطالعه قرار می‌دهد. نتیجه‌گیری نیز در بخش ۵ بیان شده است.

۲- مروری بر روش‌های شباهت‌سنجی توزیعی

روش‌های شباهت‌سنجی توزیعی مبتنی بر این فرضیه هستند که «دو واژه هدف از لحاظ توزیع به هم نزدیک و از لحاظ معنی مشابه هستند اگر آن دو تعداد زیادی واژه هم‌رخداد مشترک داشته باشند که از لحاظ نحوی رابطه یکسانی با هر دو واژه هدف دارد». از این رو اگر چه روش‌های شباهت‌سنجی توزیعی تفاوت‌های زیادی دارند، اما روند زیر تقریباً در همه آن‌ها طی می‌شود:

- انتخاب واحد شباهت‌سنجی مثل واژه یا ترکیب واژه و نقشش در جمله؛
- شمارش واحدهای هم‌رخداد در یک پنجره مشخص در متن؛
- اندازه‌گیری درجه ارتباط^۲ بین واحدهای هم‌رخداد با واژگان هدف که با $SoA(w, x)$ نشان می‌دهیم؛
- نمایش واژگان هدف توسط بردار یا تابع توزیع در فضای هم‌رخدادی و تلاش در جهت کاهش ابعاد؛ فضای هم‌رخدادی فضایی با ابعادی برابر با تعداد واحدهای هم‌رخداد است؛ مثلاً اگر واحد را واژه انتخاب کنیم و مجموعه واژگان متفاوت V باشد، فضای هم‌رخدادی، $|V|$ بعدی خواهد بود.
- محاسبه فاصله بین بردارهای هدف یا میزان شباهت توابع توزیع با کمک معیارهای توزیعی مناسب؛

معیارهایی که در این مقاله بررسی می‌شوند، در سه گروه معیارهای فضایی، معیارهای مبتنی بر اطلاعات متقابل نقطه‌ای و معیارهای مبتنی بر آنتروپی طبقه‌بندی می‌شوند که در ادامه به آن‌ها می‌پردازیم.

۲-۱ معیارهای فضایی

گروه معیارهای فضایی واژگان هدف را به صورت برداری نمایش

تعیین سنس واژه^۱، خلاصه‌سازی خودکار متن، دسته‌بندی اسناد و استخراج اطلاعات از متن ضروری است.

منظور از شباهت‌سنجی واژه‌ها، اندازه‌گیری میزان نزدیکی یا دوری معنای دو واژه است. برای این منظور روش‌های متفاوتی وجود دارد. در روش‌های مبتنی بر پایگاه دانش واژه‌های معنایی مرتبط با یک واژه خاص در یک شبکه معنایی لیست می‌شوند. در این روش‌ها واژه‌های متن به عنوان مفاهیم شناخته می‌شوند که این مفاهیم به طور معمول در شبکه‌های معنایی نظیر وردنت [۷] موجودند. وردنت از ساختار گراف جهت‌دار بدون دور برای نمایش مفاهیم و ارتباطشان استفاده می‌کند. براساس همین ساختار، برخی از روش‌های شباهت‌سنجی مبتنی بر وردنت از ویژگی یال‌ها و برخی از ویژگی گره‌ها استفاده می‌کنند.

رویکرد دیگر در شباهت‌سنجی معنایی واژگان روش‌های توزیعی هستند. در روش‌های توزیعی تفاوت محتوای اطلاعاتی دو واژه هدف w_1 و w_2 ، به صورت تابعی از احتمال حضور واژگان ظاهر شده مجاور با واژه w_1 در جملاتی با حضور واژه w_2 و برعکس، در انبوهی از اسناد مورد بررسی، اندازه‌گیری می‌شود. این دسته از روش‌ها در مواردی که به شبکه‌های معنایی دسترسی نداریم هم قابل استفاده‌اند؛ ولی بعضاً با مشکلاتی چون تلفیق سنس‌های مختلف یک واژه دست به گریبان هستند. رویکرد سوم، ترکیبی از دو رویکرد اول را در خود دارد و شباهت معنایی مفاهیم را با استفاده از توزیع واژگان در گروه‌های مختلف که مفاهیم درشت‌دانه هستند، محاسبه می‌کند. مفاهیم درشت‌دانه از شبکه‌های معنایی نظیر وردنت استخراج می‌شوند؛ در عین حال برای محاسبه توزیع واژگان از پیکره متنی استفاده می‌شود. لذا به هر دو گروه منابع احتیاج دارد.

در زبان انگلیسی تلاش‌های خوبی برای سنجش قدرت روش‌های مختلف شباهت‌سنجی معنایی واژگان و مقایسه آن‌ها انجام شده است [۱۵، ۵، ۲، ۳]. در حالی که در زبان فارسی چنین بررسی به صورت بسیار محدود صورت گرفته است. از جمله ضرورت‌های انجام چنین ارزیابی این است که معیارهای فوق غالباً برای حل یک مسئله بزرگ‌تر مثل ساخت وردنت [۱۰] و یا استخراج اطلاعات [۱۱، ۱۲] استفاده می‌شوند. حل مسئله بزرگ با تک تک معیارهای فوق و مقایسه نتایج حاصل کار معقولی نیست.

این پژوهش سه هدف را دنبال می‌کند. اول سنجش قدرت هر یک از معیارهای شباهت‌سنجی واژگان در زبان فارسی است. دوم مقایسه معیارهای مختلف از جنبه‌های مختلف نظیر میزان هم‌پوشانی معیارها و یا توانایی هر معیار در شباهت‌سنجی واژگان با برچسب ادات سخن مختلف است. سومین و مهم‌ترین هدف، ارائه معیارهای ترکیبی برای بهبود دقت شباهت‌سنجی در زبان فارسی است. این پژوهش گروه روش‌های شباهت‌سنجی توزیعی را انتخاب کرده است، زیرا روش‌های مبتنی بر پایگاه دانش عموماً شباهت واژه-سنس‌ها را ارزیابی می‌کنند و نه شباهت واژگان که خود مستلزم استفاده از تکنیک‌های تشخیص سنس است که ما به دنبال آن نیستیم.

2- strengths of association

1- sense

۲-۲ معیارهای مبتنی بر اطلاعات متقابل نقطه‌ای

روش PMI بر مبنای باهم‌آیی واژگان ارائه شده است. هرچه دو واژه بیشتر با هم ظاهر شوند شباهت یا ارتباط معنایی بیشتری برایشان برآورد می‌شود. میزان ارتباط معنایی دو واژه هدف بر طبق معیار PMI به صورت زیر است:

$$PMI(w_1, w_2) = \log \frac{p(w_1 \wedge w_2)}{p(w_1)p(w_2)}, \quad (4)$$

در این رابطه $p(w_1 \wedge w_2)$ احتمال باهم‌آیی دو واژه w_1 و w_2 در یک پیکره بزرگ است. همچنین $p(w_1)$ و $p(w_2)$ احتمال رخداد واژه w_1 و w_2 در پیکره است.

۲-۳ معیارهای مبتنی بر آنتروپی

در این گروه از روش‌ها برای واژگان هدف در فضای هم‌رخدادی یک تابع توزیع فرض می‌شود. به عبارت دیگر احتمال هم‌رخدادی واژگان هدف بر حسب واژگان همسایه محاسبه می‌شود و سپس میزان شباهت توابع توزیع تعیین می‌شود.

۲-۳-۱ معیار KLD

در تعریف معیار KLD^۳ برای شباهت‌سنجی واژگان از تابع توزیع واژگان هدف بر حسب واژگان مرتبط همسایه استفاده می‌شود. شباهت دو تابع توزیع احتمال نیز به کمک آنتروپی نسبی تعیین می‌شود. با داشتن دو تابع جرم احتمال $p(x)$ و $q(x)$ ، آنتروپی نسبی آن‌ها به طریق زیر محاسبه می‌شود:

$$D(p \parallel q) = \sum_{x \in X} p(x) \log \frac{p(x)}{q(x)} \quad q(x) \neq 0. \quad (5)$$

به‌طور شهودی، اگر $p(x)$ تابع جرم احتمال متغیر تصادفی X باشد، $D(p \parallel q)$ نشانگر اطلاعات از دست رفته در تخمین $p(x)$ با $q(x)$ خواهد بود. به بیان دیگر، $D(p \parallel q)$ نشان‌دهنده تفاوت بین دو توزیع است. اکنون با داشتن توابع توزیع واژگان هدف بر حسب واژگان هم‌رخدادی معیار KLD به صورت زیر تعریف می‌شود:

$$KLD(w, w_r) = D(d_1, d_r) = \sum_{w \in F} P(w | w_1) \log \frac{P(w | w_1)}{P(w | w_r)} \\ = \sum_{w \in C(w_1) \cup C(w_r)} p(w_1 | w) \log \frac{P(w | w_1)}{P(w | w_r)} \quad (6)$$

در این رابطه d_1 و d_2 نشان‌دهنده تابع توزیع واژگان هدف و V مجموعه کل واژگان پیکره است.

۲-۳-۲ معیار JSD

معیار KLD یک معیار متقارن نیست و میزان شباهت w_1 به w_2 ممکن است متفاوت با میزان شباهت w_2 به w_1 باشد. برای حل

می‌دهند و سپس تلاش می‌کنند که فاصله بردارها را اندازه‌گیری کنند. شکل ۱ نمایش واژگان هدف بر حسب واژگان هم‌رخدادی با آن را به صورت برداری نشان می‌دهد. چالشی که این گروه با آن مواجه هستند، نحوه اندازه‌گیری میزان ارتباط واژگان هم‌رخدادی با واژگان هدف است. در ادامه سه معیار فاصله کسینوسی، اقلیدسی و منتهن را مرور می‌کنیم.

۲-۱-۱ کسینوس

معیار کسینوس از اولین و پر استفاده‌ترین معیارهای شباهت‌سنجی است. با داشتن دو واژه w_1 و w_2 ، معیار کسینوس درواقع کسینوس زاویه بین بردار $\vec{w}_1$ و بردار $\vec{w}_2$ را محاسبه می‌کند. اگر تعداد واژگانی که هم با w_1 و هم با w_2 هم‌رخدادی هستند زیاد باشد، آنگاه زاویه بین بردارهای $\vec{w}_1$ و $\vec{w}_2$ کوچک خواهد بود و کسینوس زاویه‌شان بزرگ خواهد شد و این نشان‌دهنده بیشتر بودن ارتباط یا شباهت معنایی w_1 و w_2 است. معیار کسینوس در بازه ۰ تا ۱ خروجی می‌دهد که ۰ نشانگر تقریباً غیر مرتبط بودن و ۱ بیانگر ترادف w_1 و w_2 است. این معیار از رابطه ۱ قابل محاسبه است:

$$Cos(w_1, w_r) = \frac{\sum_{w \in C(w_1) \cap C(w_r)} (p(w | w_1) \times p(w | w_r))}{\sqrt{\sum_{w \in C(w_1)} p(w | w_1)^2} \times \sqrt{\sum_{w \in C(w_r)} p(w | w_r)^2}} \quad (1)$$

در عبارت فوق $C(t)$ ، مجموعه واژگانی است که با واژه t در پنجره معینی در پیکره ظاهر شده‌اند. در فرمول فوق از احتمال شرطی واژه هم‌رخدادی با واژه هدف، به‌عنوان معیار اندازه‌گیری درجه ارتباط (SoA) استفاده می‌شود.

۲-۱-۲ فاصله اقلیدسی و منتهن

فاصله بین دو نقطه در فضای برداری با فرمول فاصله منتهن و فاصله اقلیدسی قابل اندازه‌گیری است. فاصله منتهن را با نرم L_1 یا به اختصار L_1 ، و فاصله اقلیدسی را با نرم L_2 و به اختصار L_2 نمایش می‌دهیم. در محاسبه فاصله منتهن، مجموع اختلافات درجه ارتباط w_1 و w_2 با هر یک از واژگان هم‌رخدادی با آن‌ها محاسبه می‌شود و در نهایت بزرگ‌تر بودن این حاصل جمع نشانگر بیشتر بودن فاصله توزیعی بین w_1 و w_2 است. فاصله اقلیدسی از جذر میانگین مربعات اختلافات درجه ارتباط w_1 و w_2 با هر یک از واژگان هم‌رخدادی با آن‌ها استفاده می‌کند. خروجی L_1 و L_2 در بازه صفر تا بی‌نهایت متغیر است، که خروجی صفر نشانگر ترادف و خروجی بی‌نهایت بیانگر عدم ارتباط معنایی است. رابطه ۲ فاصله منتهن و رابطه ۳ فاصله اقلیدسی را نشان می‌دهد.

$$L_1(w_1, w_r) = \sum_{w \in C(w_1) \cup C(w_r)} |p(w | w_1) - p(w | w_r)| \quad (2)$$

$$L_r(w_1, w_r) = \sqrt{\sum_{w \in C(w_1) \cup C(w_r)} (p(w | w_1) - p(w | w_r))^2} \quad (3)$$

3-Kullback-Leibler Divergence

سوال مورد استفاده در [۲] به همراه ۶۰ سوال چهارگزینه‌ای دیگر و با پیکره‌ای به حجم یک ترابایت به دقت ۸۱/۲۵٪ برای PMI رسیدند [۱۴] Bullinaria و Levy نیز در ۲۰۰۶ با استفاده از سوالات تافل و با هدف مقایسه میان دو روش تحلیل معنایی پنهان و اطلاعات متقابل نقطه‌ای و با یکسان گرفتن حجم و کیفیت پیکره مورد استفاده، برتری روش PMI بر LSA را نشان دادند [۴].

از جمله کارهای انجام شده در حوزه شباهت‌سنجی معنایی در زبان فارسی، می‌توان به دقت‌سنجی معیار PMI در تشخیص شباهت معنایی واژگان در سوالات آزمون تافل توسط رضایی‌پور و خسروی‌زاده [۱۶] و رسیدن به دقت ۶۰ درصدی و استفاده از روش دسته‌بندی فازی جهت تشابه‌سنجی بین جملات فارسی توسط امیرشهاب شهابی [۱۳] اشاره کرد. در پژوهش [۱۶] برای محاسبه هم‌خدادی واژگان از اسناد پیکره هم‌شهری استفاده نشده است، بلکه واژه مورد نظر در گوگل جستجو شده و از اسناد نتیجه برای محاسبه هم‌خدادی‌ها استفاده شده است.

۳-۲ پیش‌پردازش

عموم معیارهای شباهت‌سنجی واژه‌ها، نیاز به دانستن تعداد رخداد واژه‌ها یا تعداد دفعات باهم‌آیی واژگان دارند. بدین منظور لازم است که در ابتدا استاندارد مناسبی برای رسم‌الخط بر تمام متن حکم‌فرما باشد. بدین منظور می‌بایست ابتدا متن به‌طور کامل خوانده شده و نویسه‌های غیرمجاز با نویسه‌های مجاز جایگزین یا حذف شوند. برای هم‌رخداد قلمدادشدن واژگان با واژه هدف یا باید پنجره با اندازه معین تعیین شود و تا شعاع آن، واژگان همسایه هم‌رخداد تلقی شوند و یا این‌که با در نظر گرفتن جمله‌ای که واژه هدف در آن به کار رفته، سایر واژگان به کار رفته در آن جمله را هم‌رخداد با واژه هدف در نظر بگیریم که روش دوم کاراتر به نظر می‌آید چرا که همه واژگان تا یک شعاع معین ممکن است به واژه هدف چندین مرتبه مربوط نباشند و یا از مرز جمله عبور کنند. در آزمایش‌های انجام شده نیز از جمله به جای پنجره استفاده شده است. واژگان با لم یکسان می‌توانند به اشکال مختلف با کلمه هدف ظاهر شوند؛ مثلاً به صورت جمع و یا همراه با «ی» نکره. ولی هدف این است که واژگان با لم یکسان به صورت یک واژه دیده شوند؛ لذا فرایند لم‌یابی واژگان قبل از شروع آزمایش انجام شده است. دسته‌آزمایشی که در آزمایش‌ها جزء واژگان هم‌رخداد به حساب آورده نشده‌اند عبارتند از: واژگان توقف، حروف اضافه، علامت‌های نشانه‌گذاری، حروف ربط، اعداد، ضمائر و واحدهای شمارش نظیر کیلوگرم، تن، مترمکعب. برای تعیین حدود جمله، یکسان‌سازی رسم‌الخط، لم‌یابی و تشخیص ادات سخن واژگان از بسته پیش‌پردازشی متون فارسی^۷ [۹] استفاده شده است.

۳-۳ نتایج

در این بخش برای سنجش عملکرد معیارهای شباهت‌سنجی معنایی، با الگوبرداری از پژوهش‌های فوق از ۱۰۰۰ آزمون چهارگزینه‌ای ترجمه‌شده

مشکل عدم تقارن، معیار JSD^۴ ارائه شده است و در آن از رابطه زیر به جای آنتروپی نسبی برای سنجش شباهت توابع توزیع واژگان هدف استفاده می‌کند:

$$JSD(w_i, w_r) = D(d_i \Pi_{\frac{1}{r}}(d_i + d_r) + D(d_r \Pi_{\frac{1}{r}}(d_i + d_r)) \quad (7)$$

برای جلوگیری از اطناب مقاله، از نوشتن رابطه کامل صرف نظر می‌شود.

۳-۳-۲ معیار ASD

معیار ASD^۵ همان معیار KLD است که با کمی تغییر به دست آمده است:

$$ASD(w_i, w_r) = \sum_{w \in C(w_i) \cup C(w_r)} p(w | w_i) \log \frac{p(w | w_i)}{\alpha p(w | w_r) + (1 - \alpha) p(w | w_i)} \quad (8)$$

در این رابطه α یک متغیر است که معمولاً مقدار ۰/۹۹ می‌گیرد.

۳- ارزیابی قدرت معیارهای شباهت‌سنجی واژگان در زبان فارسی

در این بخش معیارهای مختلف سنجش شباهت معنایی که در بخش قبلی به مرور آن‌ها پرداختیم، مورد آزمون قرار گرفته‌اند تا دقت و عملکرد هر کدام مشخص شود.

۳-۱ دادگان

برای تشابه‌یابی به روش توزیعی لازم است که یک منبع متنی بزرگ و نسبتاً جامع از نظر موضوعی در اختیار داشته باشیم که دامنه واژگان به کار رفته در آن وسیع بوده و ضمناً هر واژه تا حد امکان در معانی مختلفش به کار رفته باشد. بدین منظور از پیکره هم‌شهری [۱] استفاده شده است که در آن اخبار روزنامه هم‌شهری از سال ۱۳۷۵ تا ۱۳۸۶ گردآوری شده‌اند. علاوه بر آن از مطالب انتشاریافته در وبگاه ایرنا هم در تکمیل پیکره فوق بهره گرفته شده است.

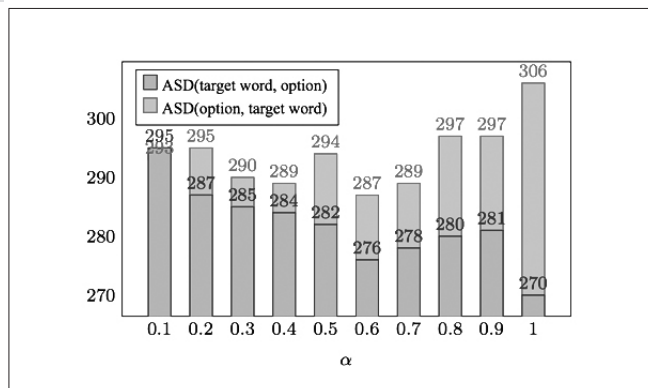
برای ساخت داده آزمون تلاش شد تا با استفاده از ترجمه سوال‌های چهارگزینه‌ای بخش واژگان آزمون‌های تافل و دکترا و تعدادی سوال تالیفی، تعداد ۱۰۰۰ سوال چهارگزینه‌ای فارسی جمع‌آوری شود. استفاده از سوال‌های تافل برای دقت‌سنجی معیارهای ارزیابی شباهت معنایی، سابقه‌ای طولانی دارد. Landauer و Dumais در ۱۹۹۷ با استفاده از ۸۰ سوال تافل، دقت روش تحلیل معنایی پنهان^۶ را در تشخیص مترادف میان واژگان زبان انگلیسی، ۶۴٪ برآورد کردند [۶] Turney نیز در ۲۰۰۱ در آزمایشی با استفاده از ۸۰ سوال تافل و ۵۰ سوال از آزمون ESL به دقت ۷۴٪ برای روش اطلاعات متقابل نقطه‌ای دست یافت [۱۵]. Terra و Clarke هم در ۲۰۰۳ با استفاده از ۱۳۰

4- Jensen-Shannon Divergence

5- α -Skew Divergence

6- latent semantic analysis

7- www.persianp.ir/toolbox.html



شکل ۳: تعداد پاسخ صحیح معیار ASD در هزار آزمون

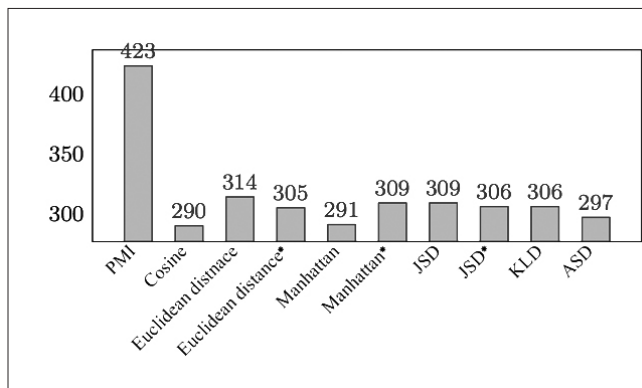
پاسخ‌های صحیح هر یک از معیارها است. هم‌پوشانی معیارهای PMI، کسینوس و فاصله اقلیدسی با یکدیگر کم است هم‌پوشانی فاصله اقلیدسی و فاصله منهن زیاد است زیرا از یک خانواده هستند و تفاوت اندکی در فرمول خود دارند. هم‌چنین معیار ASD در بهترین حالت ($\alpha = 1$) همان معیار KLD است و لذا سه معیار مبتنی بر آن‌روپی هم شبیه هم عمل می‌کنند.

۳-۳-۱ دقت معیار PMI در جمله

معیار PMI بر این اساس کار می‌کند که میزان باهم‌آیی واژگان هدف را می‌سنجد. اما واژگانی که شباهت معنایی زیادی دارند و یا مترادف هستند، معمولاً به جای هم ظاهر می‌شوند نه این که با هم ظاهر شوند. لذا در این آزمایش ۵۰۰ سوال چهارگزینه‌ای از آزمون اول انتخاب شدند و هر واژه آزمون در جمله مربوطه خود در نظر گرفته شد. سپس جمع امتیاز PMI واژه هر گزینه با تمام واژگان جمله از جمله واژه هدف محاسبه شد. چرا که گاه واژگان مترادف با یکدیگر نیز در جمله ظاهر می‌شوند و هم‌چنین همه آزمون‌ها رابطه ترادف را مدنظر نداشتند. در نهایت گزینه‌ای که بیشترین مجموع امتیاز را کسب کرده است به‌عنوان پاسخ صحیح انتخاب شده است. علیرغم این که انتظار می‌رود شباهت‌سنجی در جمله نتیجه بهتری دهد، معیار PMI تعداد ۱۴۴ سوال از ۵۰۰ سوال را صحیح پاسخ داد که دقت آن معادل ۲۹٪ است و نسبت به آزمون اول که ۴۲٪ بود، افت داشته است.

۴- ارائه معیار جدید شباهت‌سنجی واژگان به روش ترکیبی

در این بخش تلاش می‌شود تا با ترکیب معیارهای مختلف معیاری بهتر از سایرین به دست آید. در آزمون‌هایی که در بخش ۳-۳ انجام شد، مشخص شد که معیار PMI بیشترین دقت را نسبت به شش معیار دیگر دارد. هم‌چنین میزان هم‌پوشانی معیارها با یکدیگر بجز معیار PMI و Cosine نسبتاً زیاد است. پس برای بهبود نتایج شباهت‌سنجی لازم است که از ترکیب معیار PMI و Cosine با بقیه معیارها استفاده شود. سه رویکرد کلی برای ترکیب معیارها پیشنهاد می‌شود:



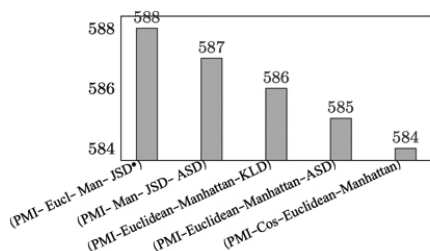
شکل ۴: تعداد پاسخ صحیح هر معیار در هزار آزمون

به فارسی استفاده شده است. در هر آزمون چهارگزینه‌ای یک واژه به‌عنوان واژه سوال و ۴ واژه در گزینه‌ها داده شده است. با استفاده از هر یک از معیارهای ذکر شده، شباهت هر یک از گزینه‌ها به واژه سوال ارزیابی شده و گزینه‌ای که شباهت معنایی بیشتری با واژه سوال داشته است به‌عنوان گزینه صحیح انتخاب شده است. شکل ۴ تعداد پاسخ صحیح هر معیار را نشان می‌دهد. مشاهده می‌شود که معیار PMI بالاترین دقت را داشته است. معیارهایی که علامت * دارند، تفاوتی که با معیار اصلی دارند این است که در رابطه آن‌ها به جای در نظر گرفتن واژه‌هایی که حداقل با یکی از دو واژه هدف، هم‌رخداد هستند، واژگانی را در نظر گرفته‌ایم که با هر دو واژه هدف هم‌رخداد هستند. به عبارت دیگر در رابطه آن‌ها به جای اجتماع از اشتراک استفاده کرده‌ایم. می‌بینیم که این تغییر رابطه منجر به بهبود معیار منهن به اندازه ۱/۸٪ شده است ولی دقت معیار JSD و فاصله اقلیدسی را کاهش داده است. همان‌طور که در بخش ۲ اشاره شد، معیار ASD متقارن نیست و در محاسبه مقدار خروجی از پارامتری به نام آلفا استفاده می‌کند. برای محاسبه دقت این معیار یکبار میزان شباهت واژه سوال به هر یک از گزینه‌ها با تغییر پارامتر آلفا در فاصله ۰ تا ۱ و بار دیگر میزان شباهت واژه‌های گزینه به واژه سوال و با تغییر آلفا در بازه ۰ تا ۱ محاسبه شده است. نتیجه حاصل در شکل ۵ نشان داده شده است. بهترین مقدار معیار ASD هنگامی است که $\alpha = 1$ که در این صورت معادل KLD است. این نمودار نشان می‌دهد که هنگام استفاده از معیار ASD بهتر است از رابطه ASD(target word, option) استفاده شود.

برای مقایسه بهتر معیارهای فوق، علاوه بر تعداد پاسخ‌های صحیح هر معیار می‌توان از روش دیگری نیز استفاده کرد. در مقایسه هر زوج معیار A و B در صورتی که تعداد پاسخ‌های صحیح معیار B بیشتر باشد، هرچه تعداد سوال‌هایی که هر دو صحیح پاسخ داده‌اند نسبت به تعداد کل پاسخ‌های صحیح معیار A بیشتر باشد، نشانگر برتری معیار B نسبت به معیار A است چرا که معیار B می‌تواند به تنهایی تشخیص‌های درست معیار A را پوشش دهد. جدول ۱ میزان هم‌پوشانی معیارهای مختلف را نشان می‌دهد. منظور از هم‌پوشانی تعداد پاسخ‌های صحیح مشترک دو معیار تقسیم بر تعداد کل

جدول ۱: هم پوشانی معیارهای مختلف

ASD	KLD	JSD	Man*	Eucl	Cos	PMI	
0.27	0.27	0.23	0.24	0.26	0.3	1	PMI
0.23	0.23	0.22	0.21	0.23	1	0.3	Cos
0.58	0.58	0.66	0.73	1	0.23	0.26	Eucl
0.35	0.45	0.63	1	0.73	0.21	0.24	Man*
0.49	0.41	1	0.63	0.66	0.22	0.23	JSD
0.98	1	0.41	0.45	0.58	0.23	0.27	KLD
1	0.98	0.49	0.35	0.58	0.23	0.27	ASD



شکل ۴: تعداد پاسخ‌های صحیح ترکیب‌های چهار تایی با اولویت به PMI

توسط PMI برنده اعلام شود و ۲) بین سه گزینه یکی به تصادف انتخاب شود. که مشاهده شد حالت اول نتایج بهتری دارد. در حالت اول، بهترین رکورد از آن سه تایی PMI-Cosin-JSD* بود که توانستند ۴۰۴ پاسخ صحیح را بیابند. بعد از این گروه، گروه PMI-Cosine-Euclidean توانستند با ۴۰۳ پاسخ صحیح به رتبه دوم دست یابند. این نتیجه کاملاً قابل انتظار بود زیرا در بخش ۳-۳ نشان دادیم که سه معیار PMI-Cosine-Euclidean هم پوشانی کمی با یکدیگر دارند و لذا ترکیب آن‌ها باید بتواند پوشش بیشتری را ایجاد کند. در حالت دوم بهترین نتیجه از آن سه معیار زیر بود: فاصله اقلیدسی، فاصله منتهن و JSD* که ۳۱۲ سوال چهارگزینه‌ای را به درستی پاسخ دادند که نسبت به معیار PMI به تنهایی کمتر است. از آن جا که هدف ما پیدا کردن بهترین ترکیب معیارها و نه مقایسه همه حالت‌های ممکن ترکیبی است، صرفاً بهترین نتیجه آورده می‌شود.

۲-۲-۴ ترکیب چهار تایی معیارها

در این بخش ترکیب‌های ۴ تایی معیارها به صورتی که یکی از اعضای آن معیار PMI باشد و اولویت نیز با پاسخ معیار PMI باشد، در مجموعه ۱۰۰۰ سوال چهارگزینه‌ای آزمایش شد. نحوه انتخاب پاسخ نهایی معیار ترکیبی چنین بود که اگر پاسخ ۳ تا از معیارها گزینه یکسانی بود، آن گزینه به عنوان پاسخ نهایی معیار ترکیبی اعلام می‌شود؛ در غیر این صورت، اگر پاسخ ۲ تا از معیارها یک گزینه و پاسخ دو معیار دیگر هم یک گزینه دیگر بود، پاسخ معیار PMI به عنوان پاسخ نهایی معیار ترکیبی اعلام می‌شود؛ در غیر این صورت، اگر پاسخ ۲ تا از معیارها گزینه یکسانی بود و دو معیار دیگر پاسخ‌های مشترکی نداشته باشند، همان گزینه مشترک به عنوان پاسخ نهایی معیار ترکیبی اعلام می‌شود؛ در غیر این صورت، یعنی حالتی که پاسخ‌های ۴ معیار همه با هم متفاوت باشند، پاسخ معیار PMI به عنوان پاسخ نهایی معیار ترکیبی اعلام می‌شود. با این ترکیب‌بندی چهارگانه‌های زیر مشترکاً رتبه اول را کسب کردند: PMI-Euclidean-Manhattan-JSD* و an-Manhattan-JSD-ASD. این دو گروه با ۵۸۸ پاسخ صحیح ۱۶۵ پاسخ صحیح بیشتر از معیار PMI به تنهایی

(۱) ترکیب بر اساس اطمینان معیارها (۲) ترکیب بر اساس رای اکثریت (۳) ترکیب بر اساس برچسب ادات سخن. در ادامه به بررسی هر یک از رویکردهای فوق می‌پردازیم.

۱-۴ ترکیب بر اساس اطمینان معیارها

برای ترکیب معیارها بر اساس اطمینان معیارها باید توجه کنیم که هر معیاری خروجی خود را در چه بازه‌ای تولید می‌کند. به عنوان نمونه معیارهای PMI و کسینوس بین ۰ تا ۱ خروجی می‌دهند و لذا نمی‌توان آن را با معیارهای دیگر که بین ۰ تا بی‌نهایت مقدار می‌گیرند به صورت مستقیم مقایسه کرد. علاوه بر مقیاس متفاوت، صفر در گروه اول به معنای شباهت کامل و در گروه دوم به معنای عدم ارتباط معنایی است. برای حل این مشکل از تابع شبه سیگموئید با رابطه $f(x) = \frac{-2}{1.1^{-x}} + 2$ استفاده شده است تا خروجی گروه دوم معیارها به ۰ تا ۱ تبدیل شود. اکنون می‌توان خروجی معیارهای مختلف را با یکدیگر ترکیب کرد. برای این منظور توابع زیر استفاده شدند: بیشینه، کمینه، میانگین و ضرب. منظور از بیشینه این است که بین مقادیری که معیارهای ترکیب‌شونده برای شباهت هر گزینه به واژه هدف اعلام می‌کنند، بیشینه گرفته شود و حاصل به عنوان میزان شباهت معیار ترکیبی برای آن گزینه در نظر گرفته شود. همین طور برای کمینه، میانگین و ضرب می‌توان تعریف نمود. در این آزمایش ۲۰۰ سوال چهارگزینه‌ای انتخاب شد و معیار PMI با معیار ASD ترکیب شد؛ نتیجه در اغلب مواقع کمتر یا مساوی دقت PMI به تنهایی (۷۷ پاسخ صحیح) بود. لذا از انجام آزمایش برای سایر معیارها صرف نظر شد.

۲-۴ ترکیب بر اساس رای اکثریت

۱-۲-۴ ترکیب سه تایی معیارها

در این بخش روش‌های مختلف ترکیب سه معیار معرفی و بررسی می‌شوند. نحوه محاسبه پاسخ ترکیبی به این صورت است که هر کدام از معیارها پاسخ مورد نظرشان را بیان می‌کنند، سپس گزینه‌ای که حداقل توسط دو معیار انتخاب شده باشد، به عنوان پاسخ معرفی می‌شود. اگر یکی از سه معیار ترکیب‌شونده PMI باشد، در حالت نبود دو رای یکسان، دو حالت زیر بررسی شدند: (۱) گزینه مشخص شده

جدول ۲: دقت معیارهای مختلف برحسب ادات سخن

ASD	KLD	JSD	Manh	Eucli	Cos	PMI	تعداد	
۰,۲۵	۰,۲۶	۰,۲۹	۰,۳۰	۰,۳۰	۰,۲۴	۰,۴۴	۵۰۴	اسم
۰,۲۹	۰,۳۰	۰,۳۰	۰,۳۲	۰,۳۵	۰,۳۲	۰,۳۶	۱۵۵	فعل
۰,۱۵	۰,۱۵	۰,۳۵	۰,۳۰	۰,۲۸	۰,۱۵	۰,۱۵	۳۹	قید
۰,۲۹	۰,۲۹	۰,۲۹	۰,۲۹	۰,۲۹	۰,۳۲	۰,۴۱	۳۰۲	صفت

collection. Knowledge-Based Systems, 22(5):382–387, 2009.

- Daniel Bär, Chris Biemann, Iryna Gurevych, and Torsten Zesch. Computing semantic textual similarity by combining multiple content similarity measures. In Proceedings of the First Joint Conference on Lexical and Computational Semantics., pages 435–440. ACL, 2012.
- William Blacoe and Mirella Lapata. A comparison of vector-based representations for semantic composition. In Proceedings of the 2012 Joint Conference on EMNLP and CNLP, pages 546–556. ACL, 2012.
- John A Bullinaria and Joseph P Levy. Extracting semantic representations from word co-occurrence statistics: A computational study. Behavior research methods, 39(3):510–526, 2007.
- James Richard Curran. From distributional to semantic similarity. 2004.
- Thomas K Landauer and Susan T Dumais. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. Psychological review, 104(2):211, 1997.
- George A Miller. Wordnet: a lexical database for english. Communications of the ACM, 38(11):39–41, 1995.
- Saif Mohammad. Measuring semantic distance using distributional profiles of concepts. PhD thesis, University of Toronto, 2008.
- Mahdi Mohseni and Behrouz Minaei-Bidgoli. A persian part-of-speech tagger based on morphological analysis. In LREC, 2010.
- Mortaza Montazery and Hesham Faili. Unsupervised learning for persian wordnet construction. In RANLP, pages 302–308, 2011.
- Alexander Panchenko and Olga Morozova. A study of hybrid similarity measures for semantic relation extraction. In Proceedings of the Workshop on Innovative Hybrid Approaches to the Processing of Textual Data, pages 10–18. ACL, 2012.
- Alexander Panchenko. Comparison of the baseline knowledge-, corpus-, and web-based similarity measures for semantic relations extraction. In Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics, pages 11–21, 2011.
- Amir Shahab Shahabi. Using fuzzy clustering to detect semantic similarity among persian sentences. Indian Journal of Fundamental and Applied Life Sciences, 5:2231–6345, 2015.
- Egidio Terra and Charles LA Clarke. Frequency estimates for statistical word similarity measures. In Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1, pages 165–172, 2003.
- Peter Turney. Mining the web for synonyms: Pmi-ir versus lsa on toefl. 2001.

۱۶. خسروی زاده، پروانه و رضاییپور، سیماء، «الگوریتم PMI (اطلاعات متقابل نقطه‌ای) معیاری جهت تشخیص شباهت معنایی کلمات»، در سومین همایش ملی زبان‌شناسی رایانشی، ۱۳۹۳.

کسب کردند که ۱۶٪ بهبود محسوب می‌شود. شکل ۶ دقت معیارهای ترکیبی چهارتایی را نشان می‌دهد.

۳-۴ ترکیب براساس برچسب ادات سخن

روش دیگر برای ترکیب معیارهای مختلف، استناد به ادات سخن است. اگر معیاری در مورد بعضی ادات سخن دقت بالاتری داشته باشد، می‌توان هنگام رای گیری به نظر آن معیار وزن بیشتری داد. برای این منظور لازم است ببینیم که هر معیار چند درصد از واژگان با ادات سخن اسم، فعل، قید و صفت را به‌طور صحیح پاسخ گفته است. جدول ۲ این موضوع را نشان می‌دهد. این جدول نشان می‌دهد که معیار PMI در شباهت‌سنجی اسم‌ها و صفت‌ها بهتر از سایرین است. هم‌چنین معیار JSD در تشخیص شباهت معنایی قیده‌ها نسبت به سایر معیارها بهتر عمل می‌کند. در مورد فعل عملکرد معیارهای مختلف تفاوت معنی‌داری ندارد. در ترکیب چهارتایی معیارها که در بخش ۴-۲-۲ به آن پرداخته شد، اولویت همیشه با معیار PMI در نظر گرفته می‌شد. لذا در آزمایش برای دخیل کردن این موضوع که معیار JSD در شباهت‌سنجی قیده‌ها بهتر عمل می‌کند، اولویت‌دهی به معیارها این گونه در نظر گرفته شد: در حالت عدم رای اکثریت، اگر برچسب ادات سخن واژه مورد آزمون قید بود، اولویت به معیار JSD داده می‌شود. از آن جا که تنها ۳۹ سوال با واژه قید در مجموعه دادگان ما وجود دارد، انتظار تغییر خیلی زیادی در نتایج نمی‌رود. در عین حال، آزمایش نشان داد که چهار معیار PMI-Manhattan-JSD-ASD با اولویت‌دهی به JSD در مورد قیده‌ها توانست به نتیجه ۵۸۲ پاسخ صحیح برسد که در نتیجه بهبودی حاصل نشده است.

۵- نتیجه‌گیری

این مقاله تلاش کرد به سه سوال پاسخ دهد: اول این که معیارهای شباهت‌سنجی توزیعی چقدر در تشخیص شباهت معنایی واژگان فارسی توانا هستند. دوم این که کدام معیارها عملکرد مشابه و کدام‌ها عملکرد مختلف دارند. و سوم آیا می‌توان با ترکیب معیارهای مختلف، دقت و فراخوانی بیشتری ایجاد کرد. آزمایش‌ها نشان دادند که PMI با ۴۲٪ دقت، در بسیاری از موارد عملکرد بهتری از سایر معیارها دارد. هم‌چنین ترکیب چهارتایی معیارها توانستند ۱۶٪ دقت PMI را بهبود بخشند.

مراجع

- Abolfazl AleAhmad, Hadi Amiri, Ehsan Darrudi, Masoud Rahgozar, and Farhad Oroumchian. Hamshahri: A standard persian text

دانشگاه صنعتی شریف، استاد و مدیری لایق، مجرب و مبدع و آموزش دانشگاهی رایانه، دانشمندی پژوهشگر و توانمند را از دست داد

سید ابراهیم ابطاحی

استادیار دانشکده مهندسی کامپیوتر - دانشگاه صنعتی شریف

پست الکترونیکی: abtahi@sharif.edu



در اولین روزهای بهار سال نو، دانشگاه صنعتی شریف ناباورانه به سوک نشست و دکتر سیدقاسم میرعمادی معاون آموزشی، استاد گرانقدر، پر تلاش و عضو خستگی ناپذیر خانواده شریف به دیار باقی شتافت.

زندگی نامه

دکتر میرعمادی دارای دکترای مهندسی کامپیوتر از دانشگاه صنعتی «چالمرز گوتنبرگ» سوئد و استاد تمام دانشکده مهندسی کامپیوتر دانشگاه صنعتی شریف بود.

ریاست و معاون آموزشی و معاون پژوهشی دانشکده مهندسی کامپیوتر، دبیر کمیته منتخب و مدیر گروه سخت افزار (معماری کامپیوتر) این دانشکده، دبیر شورای دانشگاه، رئیس کمیته رباتیک دانشگاه، سردبیر مجله تخصصی ساینتیا در «علوم و مهندسی کامپیوتر»، رئیس سیزدهمین کنفرانس بین المللی کامپیوتر ایران،

دبیر اجرایی سومین کنفرانس «آموزش مهندسی»، مسئولیت راه اندازی دوره های بین المللی دانشگاه در جزیره کیش، از جمله سوابق حرفه ای دکتر میرعمادی است. دکتر میرعمادی طی سال های ۸۹ و ۹۱، دو سال معاون آموزشی و تحصیلات تکمیلی دانشگاه صنعتی شریف بود. وی همچنین از شهریورماه سال ۹۳ تا زمان حیات مجدداً در سمت معاونت آموزشی دانشگاه صنعتی شریف در حال خدمت صادقانه به این دانشگاه بود. از جمله تالیفات وی طی ۱۰ سال اخیر، چاپ بیش از ۳۰ مقاله در مجلات، بیش از ۱۳۰ مقاله در کنفرانس های معتبر بین المللی و ۱۲ مقاله پژوهشی در کنفرانس های داخلی بودند.

راهشان پر رهرو باد. اما روزهای ورزش که ثمره وجود چند وجهی شما بود را بی دیدار چهره شاداب و تهی از خستگی شما، چه کنیم. شما که همه را همیشه می دیدید. بر من واجب است شهادت دهم بر ظمی را که از دیگران بر من - و حتماً دیگرانی - رفته بود و شما - بی تقاضای مظلومان - تا به شکلی - حتی با صبر سال ها - جبرانش نکردید از پای ننشستید. این درسی انسان ساز برای همه بود. روحتان قرین رحمت که نکونامان نامیرایند.

این استاد گرانمایه همچنین راهنمایی پایان نامه های ۷ دانش آموخته در مقطع دکتری، ۷۰ دانش آموخته در مقطع کارشناسی ارشد و ۴۰ دانش آموخته در مقطع کارشناسی را برعهده داشته است، ضمن آن که راه اندازی دو آزمایشگاه پژوهشی و یک آزمایشگاه آموزشی برای مقطع کارشناسی را در کارنامه علمی خود ثبت کرده است.

آخرین نامه به بزرگی که از بین ما رفت و در قلب ما ماند^۱

دکتر میرعمادی عزیز، دریغ که دیدار شما، برای گفتگو در مسائل کاری، که در آخرین نامه ها به برخی از همکاران از جمله من، قرار گذاشته بودید، موکول به قیامت شد و شما سید بزرگوار با چهره گشاده و روی سپید به دیدار معبود شتافتید. ما ماندیم تا ناباورانه دوره کنیم هر روز را و هنوز را تا اگر از اعمالمان در قبال شما، عبرت گرفتیم، بتوانیم با روی سپید، به دیدار شما بیاییم. این کمینه دین ما به شماست که مرور کنیم، شما برای ما چه کردید و ما با شما چه کردیم.

هر روز، خروسخوان را به شام دوختید تا در اطاق کارتان با صبر و لبخند، پاسخگوی انتظارات یا شکایت از کوتاهی هائی باشید که پاسخ را نه شما، بلکه دیگران و گاه جامعه، باید می گفت. دل آزارانه تر این که، جبرانی که طلب می شد، عموماً خواسته ای فردی و گاه دور از مرز قانون و اخلاق بود، که خط قرمز شما بود. هر دو این ها، موجب رنج و آزرده گی قلبی شما می شد. عمری غم فرزندان ما و خود را خوردید و گروهی از ما که منفعت طلبانه، فراقانون را حق می نمودیم بر شما تازیانه روحی زدیم و شما ماندید و این همه قدرشناسی فرزندان که غمخوارشان بودید و هزاران دیگر که بماند. ما با شما چه کردیم، احتمالاً به پاداش این رنج مستمر و بی پایان، تن شریفتان را سهل انگارانه بر زمین کوبیدیم، حق پناه شما شد و ماندید، بزرگوارانه و شادمانه برگشتید. ما بهانه جویانه، واقعیت این قصور را پنهان کردیم و شما ماندید و این حیرت، که با من که چنین می کنند، وای بر دیگران. بر دل گرفتید و لب نگشودید و این رنج مضاعفی بر روح زخم خورده شما شد.

و حالا ما مانده ایم و این همه فکر نواورانه و طرح اصلاحی ناتمام، که معلوم نیست با چه همت و لیاقتی می توان آن ها را به سرانجام رساند: از طرح دانشکده در دست تاسیس فناوری های همگرا گرفته، تا فکر برپائی اطاق های سرمایه های فکری برای اساتید بازنشسته دانشگاه، از شکل دهی به پژوهش در آموزش تا ارزشیابی اثر بخشی آموزش ها و هفته آموزشی که در راه است و دریغا، جلسه دوم کمیته برگزاری آن مصادف شده است با روز تشییع پیکر مطهر شما، گوئی دست سرنوشت، گل می چیند.

اما پیشینه می گوید، دانشگاه ما میراث چون شما اساتید بزرگی است که جان شیفته خدمت خود را، بی منت، چراغ نسل آینده کردند.

۱- این سوک نامه عیناً در شماره ۷۱۲ روزنامه شریف در تاریخ ۲۶ فروردین ۱۳۹۶ چاپ گردیده است

اعضاء هیئت مدیره انجمن:

آقای ابراهیم نقیبزاده مشایخ (رئیس)

آقای دکتر اسلام ناظمی (نایب رئیس)

آقای دکتر علیرضا باقری (دبیر)

آقای مهندس محمدحسن محوری (خزانه دار)

آقای دکتر محمد گنج تابش

آقای دکتر علی فرخی (عضو علی البدل)

آقای مهندس سعید امامی (عضو علی البدل)

کمیته های انجمن:

آقای ابراهیم نقیبزاده مشایخ (سرپرست کمیته انتشارات)

آقای دکتر محمد گنج تابش (سرپرست کمیته آموزش)

آقای مهندس سعید امامی (سرپرست کمیته عضویت و

روابط عمومی)

آقای مهندس علیرضا ماهیار (سرپرست کمیته

گردهمایی های علمی)

www.isi.org.ir

برای دریافت
اسلاید سخنرانی های انجمن
به وبگاه انجمن
مراجعه کنید.

www.isi.org.ir

معرفی استاندارد ۲۰۲۴۶: بازنگری‌های محصول کاری

سیدعلی آذرکار

شرکت مهندسی پدیدپرداز

کمیسیون استاندارد و تدوین مقررات، سازمان نظام صنفی رایانه‌ای استان تهران

پست الکترونیکی: ali.azarkar@pdpsoft.com

۱- مقدمه

یکی از موضوعات مهم در تضمین کیفیت نرم‌افزار، بازنگری محصولات کاری است که حین اجرای پروژه‌های نرم‌افزاری تولید می‌شود. شناسایی سطح کیفی محصولات کاری و انجام اقدامات لازم برای تصحیح و رفع زودهنگام آن‌ها، یکی از عواملی است که می‌تواند در هزینه اجرای پروژه‌های نرم‌افزاری صرفه‌جویی کرده و حتی از شکست آن‌ها جلوگیری کند.

طی سال‌هایی که از عمر مهندسی نرم‌افزار می‌گذرد، تجارب برآمده از اجرای پروژه‌های مختلف، ابزارها و روش‌هایی متنوع و گوناگونی را برای بازنگری محصولات کاری پروژه‌های نرم‌افزاری ابداع و مدون کرده است. پیچیدگی‌های ذاتی محصولات کاری و نیز نیاز به تدوین یک چارچوب عام برای بازنگری‌های مختلف انواع محصولات کاری، منجر به تدوین استاندارد ISO/IEC 20246:2017 (با عنوان «مهندسی سامانه‌ها و نرم‌افزار- بازنگری‌های محصولات کاری») شد، که در این گزارش به اختصار «استاندارد» نامیده می‌شود.

۲- مشخصات

مشخصات این استاندارد در جدول شماره ۱ نشان داده شده است.

۳- ساختار

ساختار این استاندارد به شرح زیر است:

• بند ۱: دامنه کاربرد

• بند ۲: انطباق

• بند ۳: مراجع الزامی

• بند ۴: اصطلاحات و تعاریف

• بند ۵: بازنگری‌های محصول کاری

• بند ۶: فرآیند بازنگری محصول کاری

• بند ۷: فنون بازنگری

• پیوست A (الزامی): مستندسازی بازنگری

• پیوست B (اطلاعاتی): مثال‌های بازنگری محصول کاری

• پیوست C (اطلاعاتی): ویژگی‌های بازنگری

• پیوست D (اطلاعاتی): انواع بازنگری

• پیوست E (اطلاعاتی): نگاشت به استاندارد IEEE 1028

• پیوست F (اطلاعاتی): انتخاب بازنگری بر اساس محصول کاری

• پیوست G (اطلاعاتی): بازنگری‌های - نگاشت به چرخه‌حیات

• پیوست H (اطلاعاتی): سنجش و بهبود بازنگری

• پیوست I (اطلاعاتی): پشتیبانی ابزاری

• کتابشناسی

۴- کلیات

بازنگری‌های محصولات کاری به عنوان ابزاری مهم برای شناسایی زودهنگام نواقص در اکثر پروژه‌های نرم‌افزاری استفاده می‌شود. از آن‌جا که رفع زودهنگام نواقص منجر به صرفه‌جویی زیادی (زمانی

جدول ۱: مشخصات استاندارد

شماره:	۲۰۲۴۶
قسمت:	-
نوع سند:	استاندارد بین‌المللی
عنوان لاتین:	Software and Systems Engineering — Work Product Reviews
عنوان فارسی:	مهندسی سامانه‌ها و نرم‌افزار - بازنگری‌های محصول کاری
تعداد صفحات:	۴۲
سال چاپ:	۲۰۱۷
زبان:	تک زبانه، انگلیسی
شماره ویرایش:	ندارد
متمم و اصلاحیه:	ندارد
استاندارد(های) جایگزین شده:	ندارد
کد رده‌بندی استاندارد (ICS):	۳۵.۰۸۰
نهاد استاندارد‌گذار:	ایزو
شماره استاندارد معادل فارسی:	-
ناشر استاندارد فارسی:	-
سال چاپ:	-
شماره ویرایش:	-
تعداد صفحات:	-

و هزینه‌ای در پروژه می‌شود، بازنگری‌ها نقش مهمی در پروژه‌ها داشته و علاوه بر آن تأثیرات زیادی بر ارتقای کیفیت محصولات کاری (و محصول نهایی) دارند. در عمل، بازنگری‌ها برای طیف گسترده‌ای از مقاصد، علاوه بر شناسایی و رفع نواقص، استفاده می‌شوند.

بازنگری‌ها به چندین روش می‌توانند رده‌بندی شوند. در این استاندارد، بازنگری‌ها در دو رده «رسمی» و «غیررسمی» قرار می‌گیرند. فنون متعددی از بازنگری می‌توانند طی این فرآیند استفاده شود، مانند بازنگری‌های مبتنی بر نقش برای بازنگری‌های انفرادی یا بازنگری‌های مبتنی بر بازبینی (چک‌لیست) در جلسات بازنگری.

فرآیند عام بازنگری که در قسمت ۵ توضیح داده شده، در برگیرنده تعدادی از ویژگی‌های قابل انتخاب (از جمله فنون بازنگری) است. این ویژگی‌ها به کاربر امکان می‌دهد تا نوع خاص بازنگری خود را، حسب شرایط ویژه پروژه، پیکربندی کند. این ویژگی‌ها در قسمت ۴-۱ تعریف شده است. پیکربندی فرآیند عام، به کاربران این انعطاف را می‌دهد که بازنگری‌ها را متناسب با مقاصد خود تعریف کرده، ضمن آن که هنوز با محدودیت‌های خود در باب کارایی و اثربخشی منطبق هستند؛ به جای آن که مجبور به انتخاب یک نوع خاص بازنگری شوند که نتوانند در عمل آن را به شکل کامل انجام دهند.

از حیث تاریخی، انواع متمایزی از بازنگری‌ها در ادبیات این موضوع وجود داشته ولی برخی از آن‌ها فقط در میزان تأکیدی که بر یک ویژگی وجود داشته، با یکدیگر متفاوت هستند. به عنوان مثال، برخی اعتقاد دارند که تفاوت بین «وارسی» (Inspection) و «بازنگری فنی» (Technical Review) خیلی ساده در این نکته است که «وارسی» نیازمند بهبود فرآیند است.

۴-۱ ویژگی‌های بازنگری

فهرستی از ویژگی‌های بازنگری که بر اساس آن می‌توان یک بازنگری را تعریف کرد، به شرح زیر است:

- هدف، که می‌تواند یکی از این موارد باشد: کشف موضوعات، ارزشیابی کیفی و ایجاد اعتماد نسبت به محصول کاری، آموزش بازنگرها، کسب اجتماع، خلق ایده‌های جدید، تشویق و توانا کردن نویسندگان برای بهبود، کاهش زمان و هزینه و مخاطره توسعه نرم‌افزار

- نقش‌ها، که می‌تواند یکی از این موارد باشد: نویسنده، مشتری، تسهیل‌گر، مدیریت، خواننده، ثبت‌کننده، رهبر بازنگری، بازنگری‌کننده، هماهنگ‌کننده بازنگری، رهبر فنی

- فنون بازنگری فردی

- فعالیت‌های اختیاری

- تعداد بازنگری‌کننده‌ها

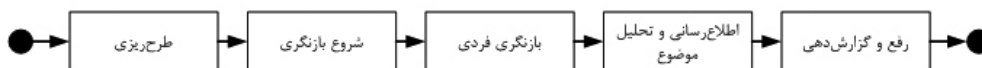
• تعداد بازنگری‌های طرح‌ریزی شده

- گزارش‌دهی رسمی، که می‌تواند یکی از این موارد باشد: ثبت سوابق موضوع، گزارش‌دهی رخداد، گزارش‌دهی بازنگری

- آموزش‌های لازم، که می‌تواند یکی از این موارد باشد: دوره آموزش برای رهبران بازنگری/تسهیل‌گرها، دوره آموزشی در مورد فرآیند بازنگری، دوره آموزشی برای مدیران درباره برنامه‌ریزی راهبردی، استفاده و تامین منابع بازنگری

- بهبود بازنگری، که می‌تواند یکی از این موارد باشد: بهبود موردی، بهبود مبتنی بر نتیجه، بهبود به عنوان یک فعالیت ویژه

- معیارهای ورود و خروج (به/از بازنگری)، که می‌تواند یکی از این موارد باشد: معیارهای ورود: اتمام بازبینی توسط نویسنده، اتمام موفقیت‌آمیز بازنگری تمامی مستندات، پایان یک بازنگری غیر رسمی، مستندسازی تمامی بازنگری‌های فردی با معیارهای خروج، اتمام مستندسازی تمامی موضوعات، رفع رضایت‌بخش موضوعات،



شکل ۱: فرآیند بازنگری محصولات کاری

- مشارکت‌کنندگان، بهبودهای بالقوه را در شیوه‌های کاری خود شناسایی کرده‌اند.

۵-۱- فعالیت‌ها و کارها

فرد یا افرادی که مسئول بازنگری محصول کاری هستند، باید فعالیت‌ها و کارهای زیر را طبق سیاست‌ها و رویه‌های سازمانی (که در قبال بازنگری محصول وجود دارد) انجام دهند. فعالیت‌هایی که در شکل شماره ۱ نشان داده شده، در ادامه توضیح داده شده است.

۵-۱-۱ طرح‌ریزی

این فعالیت شامل کارهای زیر است:

- محدوده بازنگری، شامل هدف، محصولی که باید بازنگری شود، مشخصه‌های کیفی که باید ارزشیابی شود، حوزه‌هایی که باید روی آن تمرکز شود، اطلاعات پشتیبان مانند استانداردها، نیروی انسانی و زمان‌بندی بازنگری باید تعریف شود.
- مشخصه‌های بازنگری باید تعریف و توافق شود.
- مشارکت‌کنندگان در بازنگری به همراه نقش (نقش‌های) مورد انتظار از آن‌ها باید شناسایی و توافق شود.

۵-۱-۲ شروع بازنگری

این فعالیت شامل کارهای زیر است:

- ملزومات بازنگری مورد نیاز باید بین مشارکت‌کنندگان توزیع شود.
- رهبر (مسئول) بازنگری باید محدوده و مشخصات بازنگری را به مشارکت‌کنندگان در بازنگری اطلاع‌رسانی کند.
- رهبر بازنگری باید نقش‌ها، مسئولیت‌ها و تمرکز بازنگری را به هر یک از مشارکت‌کنندگان اطلاع‌رسانی کند.
- نویسنده (یا یک فرد ذی‌صلاح مناسب) ممکن است محصول کاری تحت بازنگری را در ابتدا توصیف کند.
- ممکن است آموزش‌های لازم به بازنگرها داده شود.

۵-۱-۳ بازنگری فردی

این فعالیت شامل کار زیر است:

- هر بازنگری‌کننده باید یک بازنگری را به منظور شناسایی موضوعات مرتبط با محصول کاری انجام دهد.

۵-۱-۴ اطلاع‌رسانی و تحلیل موضوع

این فعالیت شامل کارهای زیر است:

رسیدن به یک اجماع، انجام تصمیم‌گیری‌های لازم، تهیه گزارش بازنگری

۴-۲ انواع بازنگری

در زیر، فهرستی از بازنگری‌های که عموماً در ادبیات مهندسی نرم‌افزار به آن اشاره شده آمده است:

- بازبینی نویسنده
- بازبینی همکار
- بازبینی گروهی غیررسمی
- بازبینی واریسی
- بازنگری نقطه عطف
- بازنگری مشترک
- بازنگری رومیزی مشترک
- بازنگری فنی
- مرور کلی

۵- فرآیند بازنگری محصول کاری

فرآیند بازنگری محصول کاری دربرگیرنده فعالیت‌هایی برای بازنگری محصول کاری است (شکل ۱). باید توجه داشت که فرآیندهای نشان داده شده در شکل ۱ همواره برای محصولات «کامل» استفاده نمی‌شود. بلکه می‌تواند بر روی «بخشی» از محصولات هم انجام شود. روشن است که در این حالت، این فعالیت‌ها نوعاً چندین مرتبه فراخوانی شده تا یک بازنگری کامل برای یک محصول کاری «کامل» انجام شود.

هدف فرآیند بازنگری محصول، ایجاد یک چارچوب ساخت یافته ولی قابل انعطاف بوده که از آن فرآیندهای بازنگری (هم رسمی و هم غیر رسمی) می‌توانند برای زمینه‌ها و مقاصد خاص مناسب‌سازی شوند.

بر اثر پیاده‌سازی و اجرای موفقیت‌آمیز یک فرآیند بازنگری محصول:

- نواقص/ موضوعات در محصول کاری شناسایی شده است.
- مشخصه‌های کیفی محصول ارزشیابی شده است.
- بازنگرها دانشی را درباره محصول کاری کسب کرده‌اند.
- اجماعی در مورد تصمیم‌های اتخاذ شده، به دست آمده است.
- ایده‌های جدیدی تولید شده است.
- بروزرسانی‌ها در محصول کاری ایجاد شده است.

۶-۱ فنون بازنگری فردی

فنون مرتبط با فعالیت «بازنگری انفرادی» که در شکل ۱ نشان داده شده، در ادامه این قسمت بحث شده است.

۶-۱-۱ بازنگری موردی

روشی بسیار رایج توسط بازنگرها برای کشف و شناسایی نواقص است. این رویکرد کاملاً ساخت‌نیافته است. به طور ساده، از هر بازنگر خواسته می‌شود تا آن‌جا که ممکن است هر نقص از هر نوع را پیدا کند؛ بدون این که کم‌ترین راهنمایی به فرد در این خصوص داده شده باشد. در این رویکرد، بازنگری‌کننده محصول کاری را صفحه به صفحه بررسی کرده و موضوعات مستندسازی و نواقص را شناسایی می‌کند.

۶-۱-۲ بازنگری مبتنی بر بازبینه

این یک رویکرد، سیستماتیک برای شناسایی نواقص مبتنی بر یک بازبینه است. اگر بازبینه‌های مختلفی تهیه و به بازنگرها داده شود، طیف گسترده‌ای از نواقص یافت شده و از مشکل یافتن نواقص مشابه که ممکن است در رویکرد موردی رخ دهد، جلوگیری می‌کند. یکی از معایب این رویکرد آن است که بازنگری‌کننده خود را محدود به بازبینه کرده و محتمل است که از سایر نواقص بالقوه غافل بماند. البته باید به بازنگری‌کننده یادآوری شود که مسئولیت آن‌ها بسیار گسترده‌تر از تکمیل بازبینه است. بازبینه‌ها شامل تعدادی سوال، مبتنی بر نواقص بالقوه بوده که از تجارب دیگر پروژه‌ها استخراج شده است. امکان دارد بازبینه‌ها برای هر محصول کاری تحت بازنگری متفاوت باشد. مثلاً، بازبینه‌ای که برای «سند نیازمندی‌ها» تهیه شده، با آن‌چه برای «سند طراحی» تدوین شده، متفاوت است. بازبینه‌ها حتی می‌توانند بر اساس روش‌گان توسعه و یا حتی حوزه کسب‌وکار نیز تفاوت داشته باشند.

۶-۱-۳ بازنگری مبتنی بر سناریو

در این رویکرد، به بازنگرها راهنماهایی ساخت‌یافته داده می‌شود که چگونه باید محصول تحت بازنگری را مطالعه کنند. وقتی نیازمندی‌ها، طراحی‌ها یا آزمون‌ها به شکل مناسبی مستند شده باشد (به عنوان نمونه در قالب کارخواست^۱)، در این صورت بازنگری‌کننده یک اجرای آزمایشی از آن‌ها، بر اساس کاربرد مورد انتظار محصول کاری، انجام خواهد داد. مخاطره مرتبط با این رویکرد آن است که بازنگری به سناریو محدود شده و ممکن است نواقصی که توسط سناریو پوشش داده نشده، هیچ‌گاه کشف و شناسایی نشوند.

۶-۱-۴ بازنگری مبتنی بر منظر(دیدگاه)

• موضوعات شناسایی شده باید اطلاع‌رسانی شود.

• موضوعاتی که قبلاً شناسایی شده و هر موضوع جدیدی که طی این فعالیت شناسایی شده، باید تحلیل شده تا بر اساس اقدام بعدی که روی آن‌ها انجام شود، وضعیتی به آن‌ها اختصاص داده شود.

• براساس وضعیت، موضوعات باید به فرد یا تیم مناسب اختصاص داده شود.

• مشخصات کیفی محصول تحت بازنگری باید ارزشیابی شده و همراه با سایر معیارهایی مرتبط، برای تصمیم‌گیری در مورد بازنگری استفاده شود.

۵-۱-۵- رفع و گزارش دهی

این فعالیت شامل کارهای زیر است:

- گزارش رخداد برای موضوعاتی که نیازمند تغییر در فرآورده‌هایی به غیر از محصول کاری است، باید ایجاد و به فرد یا تیم اختصاص یافته اطلاع‌رسانی شود.
- برای موضوعاتی با وضعیتی که نیازمند تغییر در محصول کاری است، باید اقدام شود.
- خاتمه اقدامات بازنگری روی محصول کاری باید تایید شود، در غیر این صورت وضعیت آن باید به‌روز شود.
- محصول کاری بازنگری شده، هنگامی که دستورالعمل تصمیم بازنگری برآورده شده، باید پذیرفته شود.

۵-۲ اقلام اطلاعاتی

این استاندارد بر مستندسازی فرآیند بازنگری تاکید داشته و تولید برخی از آن‌ها را الزامی دانسته است. (الزامی بودن یک موضوع در استاندارد به این معنی است که برای تایید انطباق با استاندارد، موضوع باید الزاماً پیاده‌سازی شده و شواهدی دال بر پیاده‌سازی قابل ارائه باشد).

به عنوان نتیجه حاصل از اجرای این فرآیند، اقلام اطلاعاتی زیر باید تولید شود:

- سابقه موضوع
 - گزارش رخداد
 - گزارش بازنگری
- هر کدام از این اقلام اطلاعاتی دارای سرفصل‌هایی است که در قسمت ۷ توضیح داده خواهد شد.

۶- فنون بازنگری

این استاندارد تعدادی از فنون بازنگری را که وابسته به فعالیت‌های مختلف تعریف شده در فرآیند عام بازنگری است، تعریف می‌کند. این شامل فنون بازنگری فردی و تحلیل موضوع است.

1. usecase

شکل ۱ به آن اشاره شد، برای تحلیل موضوعاتی که توسط بازننگری انفرادی یا جلسات بازننگری ایجاد شده، استفاده می‌شود. این فنون در ادامه به اجمال توضیح داده شده است.

۶-۲-۱ تحلیل انفرادی

تحلیل انفرادی معمولاً بر روی خروجی‌هایی انجام شده که طی فعالیت «بازنگری انفرادی» تولید می‌شود. در ابتدا، تمامی موضوعات از بازننگری‌کنندگان مستقل گردآوری شده، سپس با یکدیگر مقایسه و مقابله شده، و در نهایت بر اساس شدت آن‌ها، اولویت‌بندی می‌شوند. موضوعات با هدف تعیین اقدامات مناسبی که باید انجام شود تا مورد توجه قرار گیرند، تحلیل می‌شوند. نتیجه تحلیل به افرادی که موضوع را اعلام کرده، اطلاع‌رسانی می‌شود.

۶-۲-۲ فنون جلسات بازننگری

فنون مرتبط با جلسات بازننگری (که می‌تواند به عنوان بخشی از فعالیت «اطلاع‌رسانی و تحلیل موضوع» رخ دهد) توصیف‌کننده روش‌های مختلفی است که یک گروه می‌تواند بازننگری را انجام دهد:

- بازننگری صفحه به صفحه
- بازننگری مبتنی بر بازبینی
- بازننگری مبتنی بر نقش
- بازننگری مبتنی بر سناریو

۶-۳-۲ تصمیم‌گیری گروهی

هنگامی که گروه نیاز به تصمیم‌گیری دارد، بهتر است با صلاحیت‌ترین افراد گروه انتخاب شده تا تصمیم‌گیری‌های گروه توسط آن‌ها انجام شود. این سریع‌ترین و اثربخش‌ترین روش است.

۷- مستندسازی بازننگری

این استاندارد، اقسام اطلاعاتی لازم را برای تولید مستندات حاصل از بازننگری در سه دسته کلی:

- سوابق موضوع
- گزارش رخداد
- گزارش بازننگری

برشمرده است. در ادامه این قسمت، سرفصل‌های پیشنهادی برای هر کدام از این سه دسته آمده است.

۷-۱ سوابق موضوع

طبق این استاندارد، این سند دارای سرفصل‌های عمده زیر است:

- مرور کلی
- اطلاعات خاص سند. این خود دارای سرفصل‌های زیر است:
- مرور کلی
- شناسه یکتای سند

بر اساس تحقیقاتی که انجام شده، این رویکرد در اغلب موارد کاراترین و اثربخش‌ترین شیوه کشف نواقص برای بازننگری‌های رسمی بوده است. در این رویکرد، هر بازننگری‌کننده نقش یک سودبر (ذی‌نفع) را بازی کرده و از منظر خود محصول کاری را بازننگری می‌کند. ایده این رویکرد آن است که اگر همه سودبران محصول کاری را مناسب دانسته و باور دارند که از منظر آنان قابل استفاده است، آن‌گاه محصول دارای کیفیت بالایی خواهد بود. نوعاً منظرها در این رویکرد، عبارت است از:

- تحلیل‌گر کسب‌وکار
- مالک کسب‌وکار
- طراح
- نگهدارنده (پشتیبان)
- بازاریابی
- عملیات
- برنامه‌نویس
- مقررات‌گذار
- آزمون‌گر
- کاربر

توجه به این نکته مهم است که در این رویکرد تعادل و توازن بین سودبران باید لحاظ شود، به عنوان نمونه، اگر محصول تحت بازننگری یک «سند نیازمندی‌ها» است، باید به طور معمول کاربر، طراح، و آزمون‌گر در بازننگری حضور پررنگ داشته باشند.

از آن‌جا که تمامی بازننگری‌کنندگان نمی‌توانند به سادگی یک نقش جدید بگیرند، یک سناریو برای این‌که این رویکرد سهولت‌استفاده‌تر شود، به کار گرفته می‌شود. این سناریو سه بخش خواهد داشت.

توصیف منظر سودبر که بازننگری‌کننده باید طی بازننگری داشته باشد.

توصیف سطح بالای محصول که سودبر انتظار دارد از محصول کاری تحت بازننگری توسعه یابد. (به عنوان مثال، آزمون‌گر ممکن است انتظار داشته باشد یک طرح آزمون مبتنی بر مشخصه نیازمندی‌ها توسعه یابد.)

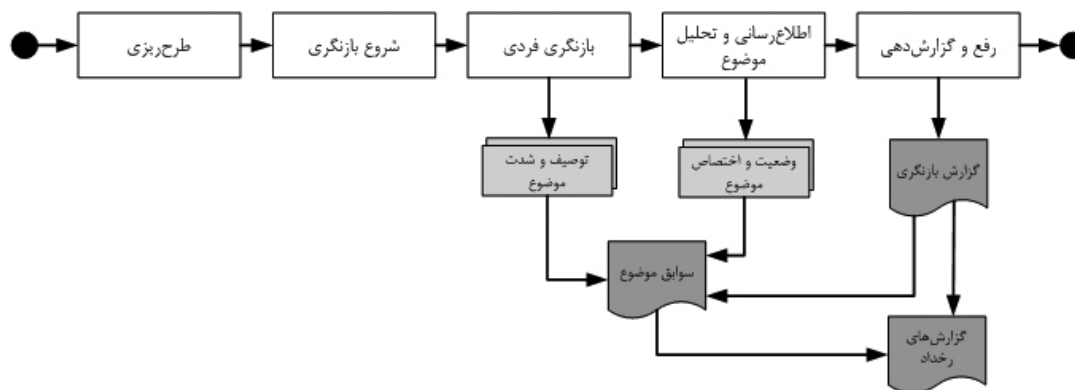
بازبینی‌ای از سوالات خاص محصول سطح بالایی است که در قسمت دوم توسعه یافته است.

۶-۱-۵ بازننگری متنی بر نقش

این یک فن بازننگری است که در آن بازننگرها محصول کاری را از منظر نقش سودبران مختلف ارزشیابی می‌کنند؛ نقش‌هایی که ممکن است با نقش روزمره آن‌ها متفاوت باشد. این نقش‌ها شبیه منظر سودبرانی است که در آن بازننگری مبتنی بر منظر استفاده شده، خواهد بود.

۶-۲ فنون تحلیل موضوع

فنون مرتبط با فعالیت «اطلاع‌رسانی و تحلیل موضوع» که پیش‌تر در



شکل ۲: نمونه‌ای از مستندسازی بازنگری

- سازمان صادرکننده (نشر دهنده)
- مرجع تاییدکننده
- سوابق تغییر
- مقدمه. این خود دارای سرفصل‌های زیر است:
- محدوده
- مراجع
- واژگان
- موضوعات. این خود دارای سرفصل‌های زیر است:
- مرور کلی
- شناسه یکتا (موضوع)
- تاریخ و زمان
- توصیف
- شدت
- وضعیت (موضوع)
- اختصاص (موضوع)
- رفع یا حل (موضوع)
- ۲-۷ گزارش رخداد
- طبق این استاندارد، این سند دارای سرفصل‌های عمده زیر است:
- مرور کلی
- اطلاعات خاص سند. این خود دارای سرفصل‌های زیر است:
- مرور کلی
- شناسه یکتای سند
- سازمان صادرکننده
- مرجع تاییدکننده
- سوابق تغییر
- مقدمه. این خود دارای سرفصل‌های زیر است:
- محدوده
- مراجع
- واژگان
- توصیف محصول کاری
- مشارکت‌کنندگان
- خلاصه یافته‌های بازنگری
- مراجع
- واژگان
- جزئیات رخداد. این خود دارای سرفصل‌های زیر است:
- اطلاعات زمانی
- اعلام‌کننده
- زمینه
- توصیف رخداد
- شدت
- اولویت
- مخاطره
- وضعیت رخداد
- ۳-۷ گزارش بازنگری
- طبق این استاندارد، این سند دارای سرفصل‌های عمده زیر است:
- مرور کلی
- اطلاعات خاص سند. این خود دارای سرفصل‌های زیر است:
- مرور کلی
- شناسه یکتای سند
- سازمان صادرکننده
- مرجع تاییدکننده
- سوابق تغییر
- مقدمه. این خود دارای سرفصل‌های زیر است:
- محدوده
- مراجع
- واژگان
- توصیف محصول کاری
- مشارکت‌کنندگان
- خلاصه یافته‌های بازنگری

- معیارهای بازنگری
- برآورد (ارزیابی) محصول کاری

برای این منظور این استاندارد، معیارهایی پیشنهاد کرده که می‌تواند طی بازنگری گردآوری شده و برای بهبود آتی آن استفاده شود:

چگالی نقص: سنجهای است برای کیفیت محصول تحت بازنگری (نحوه پراکندگی خطاها در سطح محصول چگونه است؟)

بروز نقص: سنجهای است برای کیفیت فرآیند توسعه (کجا نقص ایجاد شده است؟ در کدام مرحله از چرخه‌حیات؟)

کشف نقص: سنجهای است برای کیفیت فرآیند توسعه و آزمون (نقص در کدام مرحله از مراحل چرخه‌حیات کشف و شناسایی شده است؟)

نرخ بازنگری: سنجهای است برای سرعت فرآیند بازنگری برای یک محصول کاری خاص (به عنوان مثال، تعداد صفحات بازنگری در هر ساعت)

۱۱- خلاصه

طی اجرای پروژه‌های نرم‌افزاری محصولات کاری مختلفی تولید شده که باید بازنگری شود. هدف از بازنگری محصولات کاری، حصول اطمینان از انطباق آن‌ها با نیازمندی‌های پروژه و ارتقای سطح کیفی آن‌ها است. تجربه‌های متعددی نشان داده که رفع زود هنگام نواقص محصولات کاری می‌تواند هزینه‌های زیادی را در یک پروژه نرم‌افزاری صرفه‌جویی کرده و حتی از شکست احتمالی آن‌ها جلوگیری کند.

این استاندارد تلاش کرده یک فرآیند عام را برای انجام بازنگری محصولات کاری تدوین کند. این فرآیند می‌تواند توسط کاربران و حسب نیازها و شرایط خاص سازمان و پروژه پیکربندی شود. استاندارد مورد بحث، این فرآیند را گام‌های زیر تعریف کرده و فعالیت‌های هر کدام را نیز برشمرده است:

- طرح‌ریزی
 - شروع بازنگری
 - بازنگری فردی
 - اطلاع‌رسانی و تحلیل موضوع
 - رفع و گزارش‌دهی
- هدف دراز مدت بازنگری محصولات، شناسایی و رفع آن‌ها نیست؛ بلکه بهبود فرآیند توسعه نرم‌افزار است. به همین دلیل این استاندارد معیارهایی را نیز برای سنجش و بهبود فرآیند بازنگری ارائه داده است.

۱۲- قدردانی

از سرکار خانم مرجان طاهری که زحمت تهیه و آماده‌سازی پیش‌نویس این مقاله را متقبل شدند، سپاسگزاری می‌کنم.

۸- نمونه ای از مستندسازی بازنگری

پیوست B استاندارد، مثالی از مستندسازی بازنگری تشریح کرده است. این مستندسازی بر اساس فرآیند عام بازنگری بوده که در شکل ۱ نشان داده شده است. مستنداتی که (در این نمونه) باید طی فرآیند بازنگری تولید شوند، در شکل شماره ۲ نشان داده شده است.

۹- محصولات موضوع بازنگری

اشاره شد که محصولات کاری متنوعی طی اجرای یک پروژه نرم‌افزاری تولید شده که محتاج بازنگری هستند. این استاندارد برخی از محصولات کاری بهتر است مشمول بازنگری شوند را به شرح زیر برمی‌شمارد:

- قرارداد
- مشخصات طراحی
- طرح توسعه
- طرح نگهداری
- متن برنامه
- درخواست برای پیشنهاد
- مشخصات نیازمندی‌ها
- طرح آزمون
- مشخصات آزمون
- راهنمای کاربری
- رویه‌های کاربری
- هر کدام از محصولات کاری فوق می‌توانند قالب‌های خاص خود را (از حیث محتوایی) داشته باشند:
- متنی (مانند زبان طبیعی، متن برنامه)
- شبه سناریو (مانند مورد کاربر، مورد آزمون)
- مبتنی بر جریان (روندنما)
- مبتنی بر اطلاعات
- مبتنی بر حالت (مانند شبکه‌های پتری‌نت)
- ریاضی (مانند VDM یا Z)
- مبتنی بر UML

۱۰- معیارهای بازنگری

اگرچه اغلب تصور می‌شود که هدف اصلی بازنگری‌ها کشف نواقص است، ولی در دراز مدت هدف باید بهبود شیوه‌های توسعه و آزمون، و به تبع آن، ارتقای کیفیت محصول باشد.

آشنایی با شبکه TOR

فرزام آقا محمدی

دانشجوی کارشناسی علوم کامپیوتر دانشگاه تهران

پست الکترونیکی: f.aghahammadi@ut.ac.ir

مقدمه

شده هستند و نمی توان به محتوای آن دست یافت اما باید به این نکته توجه داشت که با این وجود هنوز نشانی مبدأ و مقصد قابل شناسایی می باشد و کاربر نمی تواند ناشناس بماند [۱].

کاربرد دیگر شبکه TOR دور زدن یا عبور از فیلترینگ می باشد. شبکه TOR با به کارگیری مسیریابی پیازی این ویژگی را پیاده سازی می کند.

این امکان را به شما می دهد که به صورت ناشناس میزبانی برای وبگاه شما باشد و رایانه شما نقش یک کارساز وب را در شبکه TOR بازی کند که فقط از طریق شبکه TOR قابل دسترسی باشد و هیچکس نتواند به هویت اصلی شما پی ببرد [۳].

تمام ترافیک در داخل شبکه TOR به صورت رمزنگاری شده است و همچنین وقتی به کمک TOR به یک کارساز متصل می شوید کارساز نمی تواند به نشانی IP شما دسترسی داشته باشد [۱].

تاریخچه

در سال ۱۹۹۵ پروژه TOR توسط ONR^۲ آغاز به کار کرد. بحث هایی در آوریل و می سال ۱۹۹۶ برای استفاده از کلیدهای DH^۳ به جای فرستادن کلید پیازی^۴ مطرح شد. اولین ارائه وانتشار رسمی

در این مقاله به معرفی شبکه TOR، کاربردهای آن، خدماتی که به ما عرضه می کند، نحوه پیاده سازی و عملکرد شبکه TOR پرداخته می شود. همچنین نکاتی که جهت استفاده درست و ایمن از این شبکه مورد نیاز است مطرح می شود. TOR یکی از ابزارهای شناخته شده جهت حفظ حریم شخصی، امنیت، عبور از فیلترینگ و فراهم کردن ناشناس بودن کاربر در اینترنت می باشد [۱،۲]. TOR کوتاه شده عبارت The Onion Router می باشد به این دلیل که برای پنهان کردن اطلاعات حساس کاربر از روش مسیریابی پیازی^۱ استفاده می کند [۷].

کاربردهای شبکه TOR

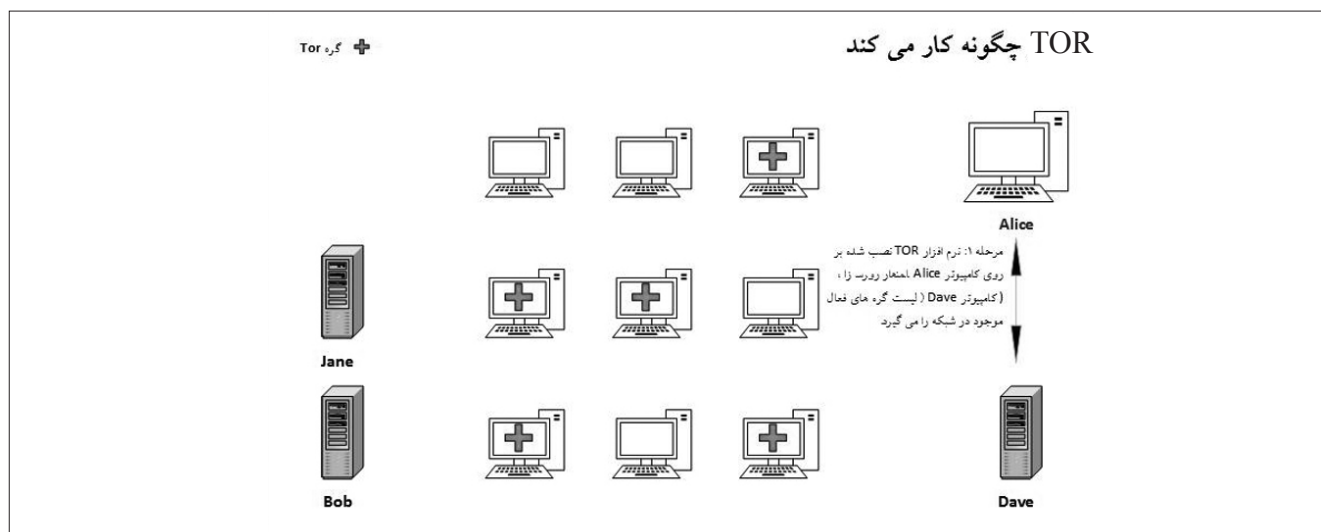
استفاده از شبکه TOR کاربر را در مقابل تحلیل و شنود ترافیک شبکه در امان نگه می دارد. در تحلیل و شنود ترافیک شبکه می توان فهمید که مبدأ و مقصد داده در شبکه کجاست و با توجه به مبدأ می توان مکان شخص مورد نظر را پیدا کرد و می توان با توجه به وبگاه هایی که بازدید می کند و اطلاعاتی که ردو بدل می کند به رفتار، علاقه مندی ها و اطلاعاتی حساس همانند نام کاربری و گذرواژه و اطلاعات بانکی شخص مورد نظر دست یافت.

شاید گمان کنید که با استفاده از پروتکل HTTPS اطلاعات رمزنگاری I- Onion Routing

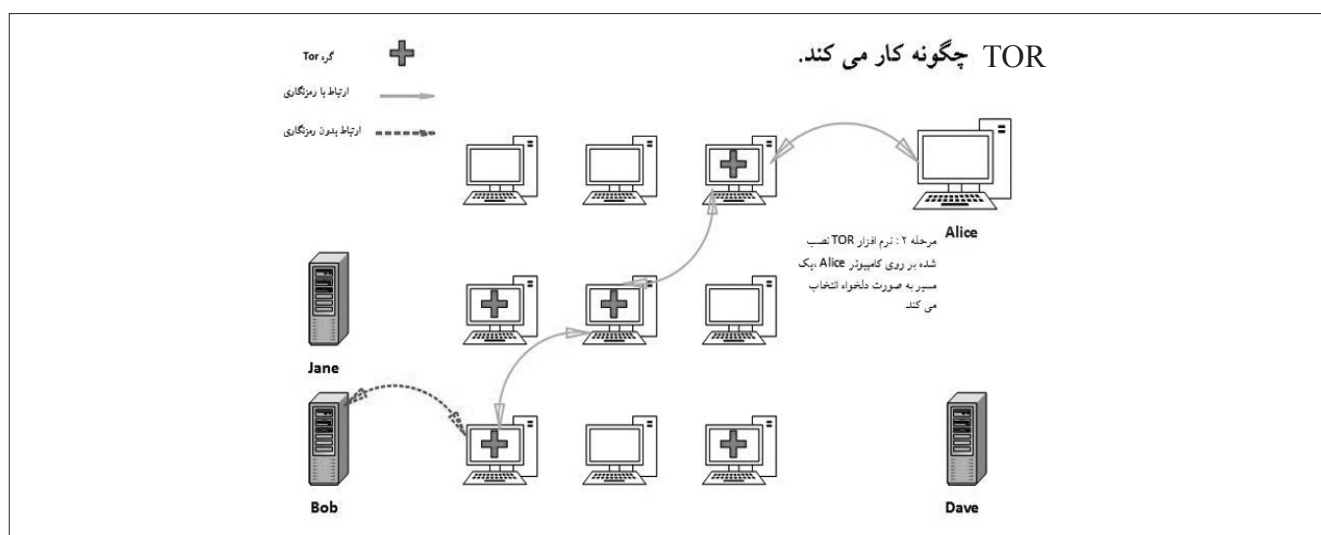
2-Office of Naval Research

3-Diffie-Hellman (DH) key

4-Onion key



شکل ۱: Alice با استفاده از نرم افزار TOR نصب شده بر روی رایانه خود به کارساز راهنما جهت گرفتن اطلاعات کارسازهای فعال در شبکه متصل می شود [۱].



شکل ۲: خط ها نشان دهنده ارتباط با رمزنگاری و نقطه چین ها نشان دهنده ارتباط بدون رمزنگاری می باشد [۱].

"Publication برای ارتباطات ناشناس و مسیریابی پیازی داده شد. در سال ۱۹۹۹ کار بر روی توسعه مسیریابی پیازی به تعویق افتاد. هیچ سرمایه گذاری جدیدی وجود نداشت. با این حال، تحقیقات و تحلیل کارها ادامه یافت. شبکه آزمایش و تحقیق نسل صفر در ژانویه سال ۲۰۰۰ بسته شد. در سال ۲۰۰۱ کار روی توسعه مسیریابی پیازی ادامه پیدا کرد که به وسیله DARPA تحت عنوان برنامه شبکه های مقاوم به خطا سرمایه گذاری شد، با هدف اولیه این که نسل ۱ را به حدی تکمیل کند که بتوان از آن در یک شبکه بتا استفاده کرد و به اهداف مقاوم بودن در برابر خطا و مدیریت منابع رسید. در سال ۲۰۰۱ جایزه "Edison Invention" برای اختراع مسیریابی پیازی داده شد. در سال ۲۰۰۲ نسل ۱ به دلیل قدیمی شدن کنار گذاشته شد کار بر روی نسل دوم آغاز شد. در سال ۲۰۰۳ سرمایه گذاری از سوی ONR برای توسعه نسل ۲، DARPA برای ساختن یک شبکه مقاوم به خطا و مدیریت منبع و برای ساختن کارسازهای پنهان انجام

مسیریابی های پیازی^۵ در اولین کارگاه پنهان سازی اطلاعات در ۳۱ می ۱۹۹۶ انجام شد. در سال ۱۹۹۷ علاوه بر پشتیبانی مالی ONR، مسیریابی پیازی توسط DARPA^۶ تحت عنوان برنامه شبکه فوق سری مورد حمایت مالی قرار گرفت. طراحی مسیریابی پیازی برای استفاده پنهان از شبکه های موبایل و برای استفاده شخصی در سایر وسایل ردیابی در کارگاه پروتکل های امنیتی در آوریل ۱۹۹۷ منتشر شد. در سال ۱۹۹۷ اکثر طراحی های نسل ۱ در IEEE منتشر شد. تغییرات بسیاری نسبت به نسل صفر به چشم می خورد. طول متغیر مسیرها، جدایی پروکسی از مسیریاب، ماژول های جداگانه رمزنگاری، بهبود عملکرد، چندین موتور جداگانه پایگاه داده برای تعداد زیاد مشترکین، قوانین خروج و غیره از جمله تغییرات مهم در طراحی بودند. در سال ۱۹۹۹ جایزه Alan Berman Research

5-Onion Router

6-Defense Advanced Research Projects Agency

گره ورودی اطلاعات را با کلید خود رمزگشایی می‌کند و سپس به گره بعدی می‌فرستد. گرهی که اطلاعات را از یک گره دریافت و به گره دیگر می‌فرستد گره انتقالی^{۱۱} یا میانی^{۱۲} نامیده می‌شود. در هر گره اطلاعات رمزگشایی شده و به گره بعدی فرستاده می‌شود. مطابق شکل ۳ در آخرین گره اطلاعات به‌طور کامل رمزگشایی می‌شود آخرین گره که اطلاعات را به مقصد می‌رساند گره خروجی^{۱۳} نامیده می‌شود.

همان‌طور که در (شکل ۳) دیده می‌شود. اطلاعات در گره خروجی بدون رمزنگاری می‌باشد. در نتیجه در مسیر بین گره خروجی و مقصد اطلاعات بدون رمزنگاری می‌باشد و در صورت تحلیل ترافیک در این مسیر اطلاعات به‌طور واضح قابل مشاهده می‌باشد. البته هنوز ناشناس بودن شما تضمین شده است چرا که هیچ کدام از گره‌ها اطلاعی از مسیر کامل ندارند. تنها اطلاعاتی که گره‌ها دارند نشانی گرهی است که اطلاعات از آن آمده و نشانی گرهی است که باید اطلاعات را برای آن بفرستد. البته اگر ارتباط با کارساز با پروتکل HTTPS باشد این مسیر هم به صورت رمزنگاری شده می‌باشد.

همچنین جهت امنیت بیشتر نرم‌افزار TOR پس از ده دقیقه از مسیر دیگری برای برقراری ارتباط استفاده می‌کند [۱،۳].

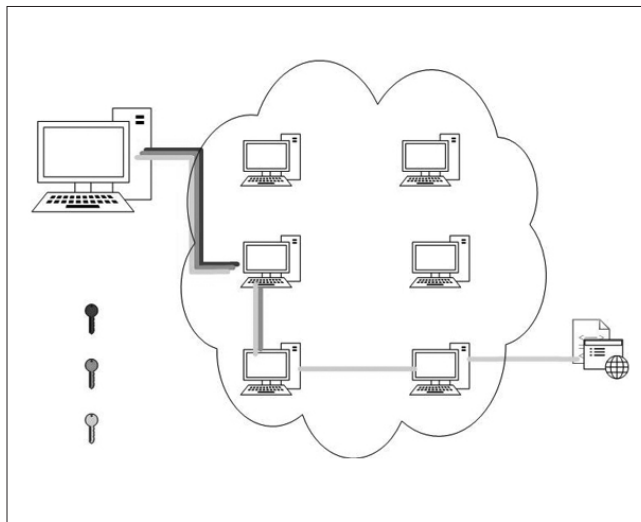
مطابق شکل ۴ می‌توان از زاویه دیگر به ساختار بسته^{۱۴}‌ها در شبکه TOR نگاه کرد. به‌طور معمول یک بسته شامل نشانی شما به عنوان فرستنده و نشانی مقصد است. اما هنگام استفاده از TOR بسته توسط چند لایه رمزنگاری می‌شود که رمزگشایی هر لایه توسط یک گره انجام می‌شود.

همچنین الگوریتم استفاده شده جهت رمزنگاری Diffie-Hellman (DH) key exchange می‌باشد [۸].

کاربران TOR

مردم عادی برای پنهان کردن اطلاعاتی نظیر وبگاه‌هایی که بازدید می‌کنند و اطلاعاتی که رد و بدل می‌کنند در مقابل ISP و هر شخص یا سازمانی که بخواهد ترافیک شبکه را تحلیل کند از TOR استفاده می‌کنند. استفاده دیگر مردم عادی جهت دور زدن فیلترینگ می‌باشد [۴]. سازمان‌های نظامی جهت ایجاد یک ارتباط امن از TOR استفاده می‌کنند. [۳]

خبرنگاران جهت محافظت از خود هنگام گزارش از جایی که دسترسی به اینترنت امن نیست از TOR استفاده می‌کنند تا موقعیتشان و اطلاعاتی که رد و بدل می‌کنند مشخص نباشد. [۳] فعالان سیاسی و حقوق بشر جهت افشای اطلاعات به صورت ناشناس در نقاطی که



شکل ۳: Tor مشتری نصب شده داده را با سه لایه رمزگذاری می‌کند، هر لایه رمزنگاری مخصوص یک گره می‌باشد [۳].

گرفت. در اکتبر سال ۲۰۰۳ شبکه TOR اجرا شد، و کد TOR به صورت آزاد و تحت مجوز ام‌آی‌تی منتشر شد. توسعه شبکه و کد توسط Tor Development Site^۷ مدیریت شد. در انتهای سال ۲۰۰۳، شبکه حدود ۱۲ گره داوطلب داشت که ۱۱ عدد در آمریکا و یک عدد در آلمان حضور داشتند. در بهار سال ۲۰۰۴ کارسازهای پنهان مکانی گسترش یافتند، مقاله طراحی TOR در USENIX Security منتشر شد، سرمایه‌گذاری از سوی EFF برای ادامه توسعه TOR آغاز شد، سرمایه‌گذاری داخلی برای کار بر روی کارسازهای پنهان مکانی ادامه یافت و در پایان سال ۲۰۰۴ بیش از ۱۰۰ گره TOR در ۳ قاره وجود داشت [۹].

مسیریابی پیازی و پیاده‌سازی آن در شبکه TOR

TOR یک مدل از پیاده‌سازی مسیریابی پیازی می‌باشد [۲]. دلیل نام گذاری مسیریابی پیازی این است که ترافیک هنگام ورود به شبکه در چند لایه رمزگذاری می‌شود و به جای این که شما به‌طور مستقیم به مقصد متصل شوید اطلاعات از چند سیستم مختلف گذرانده می‌شود و سپس اطلاعات به مقصد می‌رسد [۱،۳].

نرم‌افزار TOR نصب شده روی رایانه مطابق شکل ۱ از کارساز راهنما^۸ (در این مثال رایانه Dave) اطلاعاتی از لیست گره^۹ (کارساز)‌های فعال موجود در شبکه TOR می‌گیرد.

مطابق شکل ۲ نرم‌افزار TOR با توجه به اطلاعات دریافت شده یک مسیر به صورت دلخواه انتخاب می‌کند. و سپس مطابق شکل ۳ داده را چند لایه رمزنگاری می‌کند و به اولین گره می‌فرستد. گره اول که اطلاعات را دریافت می‌کند گره ورودی^{۱۰} نام دارد. مطابق شکل ۳

11-Transit Node

12-Middle Node

13-Exit Node

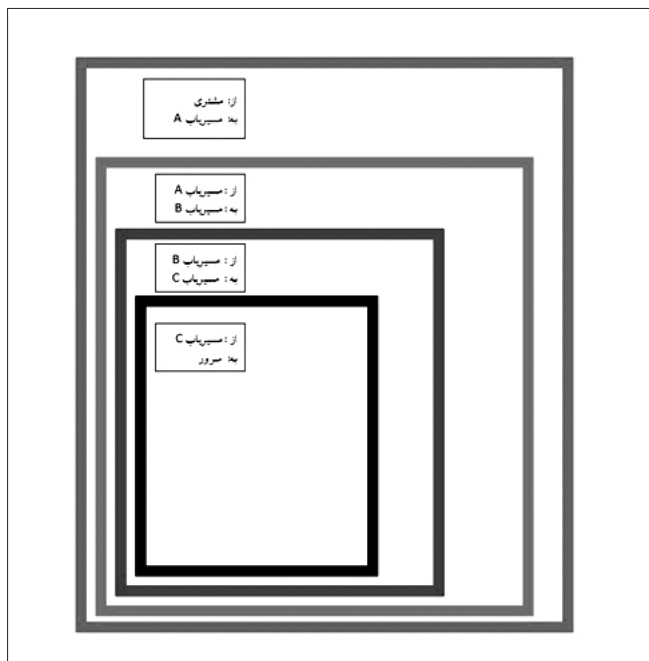
14-Packet

7-<https://www.torproject.org/>

8-Directory Server

9-node

10-Entry Node



شکل ۴: ساختار بسته‌ها در شبکه TOR [۷].

عدم استفاده از تورنت^{۱۶} هنگام استفاده از TOR

در برنامه اشتراک فایل تورنت به‌طور واضح مشاهده شده که تنظیمات پروکسی را نادیده می‌گیرد و یک ارتباط مستقیم ایجاد می‌کند و نشانی IP شما را فاش می‌کند.

عدم استفاده از افزایه^{۱۷} های مرورگر

مرورگر TOR تمام افزایه‌هایی همانند Flash, RealPlayer, QuickTime و غیره را بسته است چرا که آن‌ها می‌توانند جهت آشکارسازی نشانی IP شما دستکاری شده باشند که این کار را با دور زدن پروکسی TOR و ایجاد ارتباط مستقیم انجام می‌دهند.

استفاده از وبگاه‌هایی با پروتکل HTTPS

همان‌طور که پیش از این گفته شد در گره خارجی و مسیر بین آن و مقصد اطلاعات بدون رمزنگاری هستند و اطلاعات در این نقاط قابل مشاهده می‌باشد. در صورت استفاده از پروتکل HTTPS این مشکل برطرف شده و اطلاعات فقط در کارساز مقصد به‌طور کامل رمزگشایی شده و واضح می‌باشد.

عدم باز کردن فایل‌های بارگیری شده از TOR زمانی که برخط هستید هنگام باز کردن فایل‌های بارگیری شده از TOR باید مراقب باشید چرا که این فایل‌ها امکان دارد حاوی منابعی اینترنتی باشند که در صورت باز کردن خارج از شبکه TOR این منابع را بارگیری کنند که در این صورت نشانی IP شما را آشکار خواهد کرد. پس در زمانی که اینترنت را قطع کرده‌اید باید فایل‌ها را باز کنید. [۶]

اینترنت امن نیست یا فیلترینگ وجود دارد از TOR استفاده می‌کنند تا هویتشان به صورت ناشناس باقی بماند. [۴]
از دیگر کاربران TOR می‌توان مجریان قانون، وب‌نویس‌ها و متخصصان فناوری اطلاعات را نام برد.

چگونه وارد شبکه TOR شویم؟

راه‌های متفاوتی برای پیوستن به شبکه TOR وجود دارد. ساده‌ترین راه و راهی که اکثر مردم انتخاب می‌کنند استفاده از مرورگر TOR می‌باشد که در نشانی اینترنتی <https://www.torproject.org/download-easy.html.en> موجود می‌باشد [۲].
همچنین می‌توان از سیستم عامل Tails نیز استفاده نمود که در آن ابزار TOR موجود می‌باشد برای دسترسی به این سیستم عامل می‌توانید به نشانی اینترنتی زیر مراجعه نمایید [۵]:
<https://tails.boum.org/>

وب عمیق و خدمات پنهان TOR

TOR این امکان را به کاربران می‌دهد که مکان و هویت خود را هنگام ارائه خدماتی همانند ارائه یک وبگاه یا سرویس پیام‌رسان در امان نگه دارد [۱۱]. TOR این امکان را می‌دهد که یک گره علاوه بر این که نقش یک کارخواه را در شبکه ایفا می‌کند نقش یک کارساز را هم در شبکه داشته باشد و سرویسی را ارائه دهد. یک کارساز در شبکه TOR توسط کارخواه‌های متصل به شبکه در دسترس است، در حالی که IP اصلی کارساز کماکان پنهان است [۳].
TOR همچنین درگاهی برای دسترسی به دنیای وب عمیق می‌باشد. وب عمیق بخش عظیمی از دنیای وب می‌باشد که توسط جویشرها شناسایی نمی‌شود [۷].

Dark Net یا Dark Web زیر مجموعه‌ای از وب عمیق می‌باشد که علاوه بر این که توسط جویشرها شناسایی نمی‌شود برای دسترسی به آن به یک سیستم احراز هویت یا یک نرم‌افزار پروکسی مختص آن نیاز است. Dark Web شامل چند شبکه مثل I2P, Freenet و TOR می‌باشد [۷].

استفاده امن از TOR

جهت استفاده امن از شبکه TOR به توصیه‌های زیر توجه کنید.

استفاده از مرورگر^{۱۵} TOR

TOR از تمام ترافیکی که از رایانه شما به بیرون می‌رود محافظت نمی‌کند بلکه فقط از ترافیکی محافظت می‌کند که شما در آن برنامه تنظیم کرده‌اید که می‌خواهید ترافیک آن از TOR عبور کند. برای جلوگیری از تنظیمات اشتباه به‌طور جدی توصیه می‌شود که از مرورگر TOR استفاده شود.

16- Torrent

17- Plugin

15- TOR Browser

امنیتی جهت جلوگیری از فاش شدن هویت اصلی بسیار ضروری می‌باشد.

منابع

- 1-<https://www.torproject.org/about/overview>, accessdate 2017/2/12.
- 2- Emin Çalışkan, Tomáš Minárik, Anna-Maria Osula : Technical and Legal Overview Of the Tor Anonymity Network, Tallinn, 2015.
- 3- Peter Loshin, Practical Anonymity: Hiding in Plain Sight Online, 1st Edition, Syngress, 2013.
- 4- <https://www.torproject.org/about/torusers.html.en>, access date 2017/2/12.
- 5- James Smith, Tor and The DarkNet : Remain Anonymous and Evade NSA Spying , Pinnacle Publishers, 2016.
- 6-<https://www.torproject.org/download/download.html.en#warning> , access date 2017/2/12
- 7-John Smith, TOR: a Dark Net Journey on How to Be Anonymous Online (TOR, Dark Net, DarkNet, Deep web, cyber security Book 0) (Volume 1) , CreateSpace Independent Publishing Platform 2017.
- 8- Abdullah A. Al Qahtani, El-Sayed M. El-Alfy, Anonymous connections based on onion routing: A review and a visualization tool, Procedia Computer Science 52 (2015) 121 – 128.
- 9-<https://www.onion-router.net/History.html>, access date 2017/2/12.
- 10- Jared Norton, TOR and The Dark Net: Learn to Avoid NSA Spying And Become Anonymous Online, 2016.
- 11-<https://www.torproject.org/docs/hidden-services.html.en>, access date 2017/2/12

نکات مهم دیگر

به‌طور جدی توصیه می‌شود که هنگام استفاده از شبکه TOR وبگاه‌های مختلف از چند شناسه (نام کاربری و گذرواژه) مختلف استفاده کنید. همچنین از نام کاربری و رمز عبوری که به شخصیت شما در فضای عمومی مربوط می‌شود استفاده نکنید. همچنین در صورت انتشار فایل در شبکه TOR اطلاعات مهم را از فراداده^{۱۸} فایل حذف نمایید. نکته دیگری که باید به آن توجه کنید غیرفعال کردن جاوا اسکریپت در مرورگر مورد استفاده می‌باشد [۵].

جمع بندی

شبکه TOR ابزاری جهت عبور از فیلترینگ و فراهم نمودن ناشناس بودن کاربر در فضای اینترنت می‌باشد. شبکه TOR با استفاده از مسیریابی پیازی به این امکان دست می‌یابد. تمام اطلاعات در داخل شبکه TOR به صورت رمز شده هستند. ارتباط با وبگاه‌هایی که از پروتکل HTTPS استفاده می‌کنند این ارتباط را امن تر می‌سازد. همچنین TOR این امکان را به شما می‌دهد که به صورت ناشناس میزبانی برای وبگاه شما باشد. بهترین و ایمن ترین راه دسترسی به شبکه TOR استفاده از مرورگر TOR می‌باشد و همچنین رعایت نکات

18- Meta data

چاپ دوم منتشر شد !

**برای کسب اطلاعات بیشتر و تهیه کتاب
با شماره تلفن زیر تماس حاصل فرمایید
۶۶۴۱۲۸۶۱ (انجمن انفورماتیک ایران)**

جیسون فرید و دیوید هاین مایرهنسون، بنیانگذاران شرکت نرم افزاری 37signals هستند. محصولات تولید شده توسط شرکت آن‌ها میلیون‌ها کاربر در سراسر جهان دارد. آن‌ها در این کتاب رازهای موفقیت شرکتشان را با شما در میان می‌گذارند. این کتاب در فهرست پرفروش‌ترین کتاب‌های روزنامه نیورکتایمز قرار داشته است.



حقوق بانکداری الکترونیکی با مدیریت فناوری اطلاعات سبز و رعایت اصول مجازی سازی و رایانش ابری

سید ابراهیم ابطاحی

استادیار دانشکده مهندسی کامپیوتر - دانشگاه صنعتی شریف

پست الکترونیکی: abtahi@sharif.edu

کار که به ازای فروش هر مگابیت اینترنت، بیاید مبلغ ۱۰ ریال ناقابل را از شرکت‌های فروشنده اخذ و به حساب درآمد اختصاصی وزارت بهداشت واریز کند تا چه شود؟ کاری کارستان: به‌منظور تکمیل و تامین تجهیزات بیمارستانی و مراکز بهداشتی و درمانی و کمک به هزینه درمان بیماران تحت پوشش نهادهای حمایتی و یا صعب‌العلاج و برای اجرای طرح پزشک خانواده، پیشگیری و پوشش کامل درمان افراد هزینه شود! کدام افراد؟ - به‌منظور پوشش افرادی که ساکن روستاها و شهرهای دارای ۵۰ هزار نفر جمعیت و پایین‌تر هستند به علاوه جامعه عشایری کشور!

بدون شک در تهیه چنین سندی، نیتی خیر مطرح بود. اما تدوین‌کنندگان آن، برخلاف بودند و به دلیل خطاهایی که در نوشتن متن سند، مرتکب شدند (برای مثال: به‌جای گیگا، «مگا» نوشته بودند) به دریافت «تمشک طلایی» نایل شدند! اگرچه همین خطای کوچک آن‌ها، به نفع میلیون‌ها کاربر اینترنت کشور تمام شد، اما پس از بیست و چند سال استفاده ملی از اینترنت، هیچ‌گاه نباید از بابت رخداد چنین خطاهایی آن هم در میان نمایندگان مجلس خوشحال بود! بلکه صادقانه باید پرسید: آیا خطاهای تدوین‌کنندگان این سند و اشتباهات مشابهی که اغلب دست‌اندرکاران اینترنت گرفتارش هستند و هستیم، نشانه نپرداختن مسوولان فاوا، متولیان

برای نوشتن معرفی کتاب این شماره، به علت تراکم کاری و احساس کم‌ثمری نوشته‌ها - با توجه به سیر نزولی ناامیدکننده شمارگان کتاب‌های منتشره - بی‌انگیزه بودم که نوشته آقای عباس پورخصالیان با عنوان «خطا و نقص اطلاعات در مجلس به نفع مردم شد! تمشک طلایی برای طراحان گرانی قیمت اینترنت» را در شماره ۷۹۷ هفته‌نامه وزین عصر ارتباط مورخ ۱۲ اسفند ۱۳۹۵ به‌یادم آمد و گفتم شاید این نوشته‌ها هم در کاهش گسل رقمی (دیجیتالی) مورد نظر نویسنده موثر باشد و سپس با انگیزه بیشتر به نوشتن اقدام کردم. اما مناسب دیدم این نوشته انگیزه‌ساز را که حاوی مطالبی جالب و تحلیلی هوشمندانه بود عیناً به‌عنوان مقدمه معرفی کتاب این شماره نقل کنم تا شاید انگیزه‌ای باشد برای دیگرانی چون من که خودخواهانه (به معنی خودخواسته!) هنوز می‌نویسند در حالی که طالبی نمی‌بینند جز سردبیر عزیزی که ادواری این یادداشت‌ها را طلب می‌کند و احتمالاً فکر می‌کند مفید یا لازم است.

بند الحاقی تبصره ۱۸ لایحه بودجه سال ۹۶ که وزارت ارتباطات و فناوری اطلاعات را موظف به کاری برخلاف میل باطنی و توانایی‌اش می‌کرد، هفته پیش از لایحه حذف شد.

این بند، وزارت فاوا را موظف به چه کاری می‌کرد؟ موظف به این

و دهک‌های کم‌درآمد رعایت کند و پیش از اجرای هر پروژه عمومی، نخست برنامه‌ای برای پوشش رایگان یا استطاعت‌پذیر خدمات دولتی (مانند خدمات دولت الکترونیکی) با هدف حمایت از حقوق معلولان، محرومان و کم‌توانان ارایه و بودجه اجرای آن را نیز تامین کند. این درحالی است که تهیه‌کنندگان سند مورد بحث می‌خواستند برای تامین بودجه اجرای پزشک خانواده از شگرد Cross-Subsidization یا «یارانه‌دهی متقابل» استفاده کنند یعنی از وزارت فاوا بگیرند و به وزارت بهداشت بدهند و وزرات فاوا هم از فروشندگان خدمات اینترنتی بگیرد، فروشندگان هم که معلوم است، چه می‌کنند! در نتیجه سیاست «هویج و چماق» (تشویق و تنبیه همزمان) اجرا و بهره‌مندی از اینترنت به زیان ذی‌نفعان گیرنده خدمات از وزارت بهداشت تمام می‌شد. این نیز تنها تناقض موجود در سند مذکور نبود. تناقض دیگر، در این نکته نهفته بود که در مناطق ذی‌نفع از خدمات پزشکی، شهرهای بسیاری بودند که از لحاظ دارا بودن زیرساخت‌های رفاهی و کیفیت زندگی برتر، هم‌ردیف کلان‌شهرهای کشورند. دلیل قرار گرفتن این شهرها در میان شهرهای مورد نظر بند الحاقی، ابهام‌آمیز بودن یا هشیارانه نبودن ملاک تمایز میان مناطق محروم از مناطق برخوردار بود.

تشتت در تعیین عنوان خبر درمیان رسانه‌ها: ابهامات مطرح‌شده در مجلس و حذف نهایی بند الحاقی مذکور باعث شد، حدود نیمی از خبرگزاری‌ها، از «مخالفت مجلس با افزایش سه برابری قیمت دسترسی به اینترنت» و دیگر خبرگزاری‌ها از «مخالفت مجلس با افزایش هشت برابری» آن، گزارش دهند.

اما ابهام بزرگ‌تر، در این نکات نهفته است که: طی چه فرایندی بند الحاقی تبصره ۱۸ لایحه بودجه ۹۶ تهیه و تدوین شده بود؟ چرا این فرایند از نخستین لحظه آغاز در مجلس تا پایان، از انظار رسانه‌ها مخفی ماند و تا تنها نتیجه‌ای منتزع از فرایند، توسط خبرگزاری خانه ملت (www.icana.ir) رسانه‌ای شد؟ چرا پیش از تدوین آن بند کذا، یا پس از آن، ولی پیش از طرح رسمی آن در مجلس، از ظرفیت‌های خود مجلس، مانند دفتر ارتباطات و فناوری‌های نوین و از ظرفیت‌های موجود در کشور، به‌خصوص وزارت ارتباطات و فناوری اطلاعات برای اخذ مشورت، استفاده نشده بود؟ اگر بموقع از ظرفیت‌های موجود در کشور استفاده می‌شد، این همه خطا و حیف و میل سرمایه‌های نظام رخ نمی‌نمود.

آیا سند مورد بحث، قابل اجرا بود؟ به دلیل این که شرکت‌های فراهم‌آور خدمات اینترنتی که پس از اجرای طرح FCP تعدادشان به چندده شرکت بالغ می‌شود، به شیوه‌های متنوعی خدمات خود را می‌فروشدند، تشخیص و تعیین میزان حجم ترافیک داده‌ای فروخته شده از سوی آن‌ها برای ناظر ثالث (وزارت ارتباطات و فناوری اطلاعات) ممکن نبود و اظهار نظر وزیر ارتباطات و فناوری اطلاعات در مورد بند الحاقی مذکور در نامه‌ای خطاب به رییس سازمان مدیریت و برنامه‌ریزی به شماره ۱/۱۱۶۶۵۷ مورخ ۹۵/۱۱/۲۶

آموزش و جامعه فاوانویسان کشور به مقوله گسل دیجیتال فزاینده نیست؟

پس بیاییم ماجرای رد سند مذکور را مرور کنیم و از درس‌های آن بیاموزیم. شرایط فروش خدمات دسترسی اینترنت در کشور، کدر است! آیا قیمت دسترسی اینترنت وابسته به سرعت است؟ آیا منظور از سرعت دسترسی، همان پهنای باند دسترسی است؟ آیا قیمت دسترسی اینترنت براساس حجم مصرفی تعیین می‌شود؟ یا در تعیین قیمت دسترسی، گاهی تلفیقی از سرعت و حجم، اعمال می‌شود؟ آیا واحد مورد محاسبه در تعیین قیمت دسترسی پرسرعت به اینترنت، «مگا بیت» است یا «گیگا بیت»؟ «بیت» است یا «بایت»؟ این واحدهای سرعت چه معنا و چه تفاوت‌هایی با یکدیگر دارند؟ اگر به تبلیغات فروش دسترسی پرسرعت به اینترنت نگاه کنید، اغلب مشاهده می‌کنید که فروشندگان نیز مبنای محاسبه قیمت دسترسی پرسرعت به اینترنت را مشخص نمی‌کنند. آیا بیان مبهم فروشندگان نیز، عمدی و آگاهانه است یا ناآگاهانه؟ ماجرای رد سند مذکور از چه قرار بود؟ روز یکشنبه هشتم اسفند ۹۵، بند الحاقی تبصره ۱۸ لایحه بودجه ۹۶ به دلیل تشتت برداشت نمایندگان از میزان گرانی آتی خدمات دسترسی پرسرعت به اینترنت، از لایحه حذف شد؛ زیرا تلقی‌های مختلف از متن سند، نمایندگان را به دو دسته تقسیم کرده بود: برخی از نمایندگان تاثیر تصویب و اجرای این بند الحاقی را بر قیمت اینترنت، سه برابر شدن قیمت‌های کنونی خدمات دسترسی اینترنت ارزیابی می‌کردند. درحالی که عده‌ای دیگر از نمایندگان، فکر می‌کردند تصویب این بند، هشت برابر شدن قیمت اینترنت را برای مصرف‌کننده نهایی در پی خواهد داشت.

در نتیجه، علی لاریجانی، رییس مجلس، پس از آن که درخواست کرد: «باید مشخص شود که با تصویب این بند الحاقی تا چه میزان، نرخ اینترنت برای مردم و مصرف‌کنندگان افزایش می‌یابد» ولی پاسخی نشنید بلکه دید، کماکان با برداشت‌های دوگانه و بسیار متفاوت نمایندگان از میزان گران شدن اینترنت مواجه است، در نهایت حذف بند الحاقی مذکور را به رأی گذاشت و اکثر نمایندگان با حذف این بند از لایحه بودجه ۹۶ موافقت کردند.

ابهامات خبر حذف بند الحاقی تبصره ۱۸ لایحه بودجه ۹۶: ابهام اصلی بند مذکور، در هسته مرکزی موضوع آن نهفته بود، یعنی آنجا که متن این بند، از تامین مالی طرح پزشک خانواده از طریق «فروش هر مگابیت اینترنت» یاد می‌کرد ولی تهیه‌کنندگان این سند، دچار کاستی نظری هم بودند: سیاست‌های عدالت‌محور «تعهد خدمات عام» یا Universal Service Obligation را احتمالاً نمی‌شناختند.

توضیح این که: براساس اصول «تعهد خدمات عام»، دولت موظف است، برای هر خدمتی که به عموم مردم ارایه می‌کند، نخست جانب عدالت را درمورد مناطق محروم، اقشار آسیب‌پذیر



عنوان کتاب: مدیریت فناوری اطلاعات

سبز

تالیف و گردآوری: رضا رادفر و مسعود

ظهرابی

ناشر: انتشارات علمی و فرهنگی

زمان نشر: ۱۳۹۵

تعداد صفحات: ۲۴۵ برگ

شمارگان: ۱۰۰۰ نسخه

شابک: ۹۷۸ - ۶۰۰ - ۴۳۶ - ۰۰۶ - ۷

قیمت: ۱۸۰۰۰ تومان

مقدمه

در مقدمه کتاب، مولفین نوشته‌اند: «در حالی که برخی از مردم و سازمان‌ها، فناوری اطلاعات را به‌عنوان بخش مهمی از مشکلات زیست محیطی قلمداد می‌کنند، ولی فناوری اطلاعات می‌تواند به‌عنوان ناجی نیز عمل کند. به عبارت دیگر، فناوری اطلاعات هم به‌عنوان راهکار و هم به‌عنوان بخشی از مشکلات زیست محیطی می‌تواند مطرح باشد. ما می‌توانیم از توان فناوری اطلاعات برای رسیدگی به مسائل زیست محیطی و همچنین ساخت و استفاده از سیستم‌های فناوری اطلاعات خود به شکل بهینه‌تر استفاده کنیم... فناوری اطلاعات سبز مجموعه‌ای از سخت‌افزار، نرم‌افزار، ابزار، راهبرد و روش‌ها، برای بهبود و تقویت محیط زیستی پایدار است... سازمان‌ها از نظر قانونی، اخلاقی و اجتماعی موظف هستند، محصولات، برنامه‌ها، خدمات و شیوه‌های فناوری اطلاعات خود را با رویکرد سبز و زیست محیطی ترویج کرده و ارتقاء دهند».

محتوای کتاب

کتاب دارای مقدمه مولفین و هشت فصل مطلب و دو پیوست، و فهرست منابع و مآخذ است. پیوست شماره یک کتاب، عنوان «یک مطلب کاربردی و مفید در باره فناوری اطلاعات سبز» و پیوست دوم، عنوان «واژه نامه فناوری اطلاعات سبز» را دارد. عناوین فصل‌های کتاب به شرح زیر است:

فصل ۱: مقدمه ای بر توسعه پایدار و برنامه کاری سبز.

فصل ۲: مروری کلی بر فناوری اطلاعات سبز.

فصل ۳: فناوری اطلاعات سبز در عمل.

فصل ۴: حاکمیت فناوری اطلاعات سبز.

فصل ۵: استاندارد ISO 14001.

فصل ۶: بازگشت سرمایه فناوری اطلاعات سبز.

فصل ۷: هفت فناوری اطلاعات سبز از منظر شرکت HP.

فصل ۸: یک نمونه مطالعه موردی در حوزه فناوری اطلاعات سبز. در فصل اول کتاب آمده است: «هراس آفرینی: مخالفان دستورکار

حاکمی از همین عدم امکان تعیین میزان حجم ترافیک داده‌ای فروخته شده از سوی سازمان تنظیم مقررات و ارتباطات رادیویی است.

اما ۱۰ ریال افزایش قیمت به ازای فروش هر یک «مگابایت اینترنت» چه معنایی داشت؟ یعنی اکنون که فروش حجمی هر یک گیگابایت اینترنت، بالغ بر ۳ هزار تومان است و از آنجا که یک گیگابایت برابر با ۸ هزار مگابایت است و بنا بود «به ازای فروش هر مگابایت اینترنت، مبلغ ۱۰ ریال از شرکت‌های فروشنده اخذ» شود، پس در صورت تصویب بند مذکور، در قبال فروش حجمی هر یک گیگابایت اینترنت ۳ هزار تومانی کنونی، مبلغ ۸ هزار تومان اضافه (!) از شرکت‌های فروشنده اخذ می‌شد که این گرانی چه بسا قیمت فروش حجمی یک گیگابایت اینترنت را به ۱۱۰۰۰ تومان (۳۰۰۰+۸۰۰۰) افزایش می‌داد یعنی به اندکی کمتر از سه برابر قیمت فروش حجمی اینترنت در حال حاضر. باید توجه داشت که اگر این خطای فاحش رخ نمی‌داد و به‌جای محاسبه «۱۰ ریال عوارض به ازای فروش هر مگابایت اینترنت»، از «۱۰ ریال به ازای فروش هر مگابایت اینترنت» ذکر می‌شد، آن بند، به رغم تناقض ذاتی‌اش با برنامه ششم توسعه و با وجود سایر ابهاماتش، به دلیل وجود اکثریت «اصلاح‌طلب» موجود در مجلس، چه بسا رأی می‌آورد!

یک فقدان بزرگ در مجلس، e-Parliament، به دلیل فقدان «پارلمان الکترونیکی» در ایران، بسیاری از مباحث مهمی که توسط نمایندگان مردم در کمیسیون‌ها و کمیته‌های مجلس مطرح می‌شوند، دیر هنگام رسانه‌ای می‌شوند. آخرین نمونه، انتشار دیر هنگام خبر طرح بند الحاقی مذکور در مجلس در لحظه «فوت» آن است! به رغم آن که کل مستندات لایحه بودجه دولت در سال ۹۶ (همچون سالیان پیش) بی‌کم و کاست در وبگاه دولت الکترونیکی ایران منتشر شد، فرایند مذاکرات نمایندگان در کمیسیون‌ها، کمیته‌ها و کارگروه‌ها، به دلیل نبود «پارلمان الکترونیکی» بموقع، به صورت زمان‌واقعی و برخاسته از رسانه‌ای نشد. طنز قضیه این بود که کل این ماجرا در حالی داشت رخ می‌داد که در برنامه ششم توسعه تاکید شده است: «قیمت اینترنت نصف می‌شود!».

و اما درس‌ها؟ درس‌ها و آموزه‌های اخلاقی این ماجرا، برای مقابله با گسل دیجیتالی، گران‌سنگ‌اند! ولی چه نهادهایی باید پیش از همه از این آموزه‌ها یاد بگیرند؟ به عقیده من، نخست، شورای عالی فضای مجازی و مرکز ملی آن به علاوه شورای اجرایی فاوا و سپس رسانه‌ها! در نهایت، مجلس نیز باید اجرای سه پروژه را جدا دنبال کند تا از کاروان پیشرفت، عقب نماند: ایجاد «کمیسیون فاوا»، «پارلمان الکترونیکی» و «قانونمندی سازی ایران برخط»!

فصل ۱: کلیات.

فصل ۲: جنبه های قراردادی بانکداری اینترنتی.

فصل ۳: حقوق و تعهدات اطراف قرارداد بانکی اینترنتی.

فصل ۴: مسئولیت مدنی و کیفری در بانکداری اینترنتی.

فصل ۵: قانون حام، دادگاه صالح، و ادله اثبات دعوی در بانکداری اینترنتی.

فصل ۶: جنبه های حقوقی مدیریت، نظارت، و بازرسی بانکداری اینترنتی.

فصل ۷: تدوین مقررات برای بانکداری اینترنتی.

فصل ۸: نتیجه گیری و ارائه راهکارهای عملی و اجرایی.

در بخش نتیجه گیری و ارائه راهکارهای عملی و اجرایی کتاب آمده است: «پرسش عمده آن است که رویکرد آتی حقوق کشورمان نسبت به موضوع مهم بانکداری اینترنتی که عملاً نوع بین المللی آن واجد اهمیت فراوان می باشد، چگونه ترسیم می شود؟ توضیح این که در سطح داخلی می توان با اندکی محدودیت ضمن توسل به بانکداری الکترونیکی، بسیاری از مزایای بانکداری را تحصیل کرد، حال آن که در سطح بین المللی، اینترنت قابلیت هائی برای کاربران و بانکداران می گشاید که با هیچ شیوه دیگری نمی توان بدانها دست یافت. در هر دو مورد، پرداخت های خرد محل عمده توجه هستند. از آن جهت که در داخل کشور به شیوه الکترونیکی قابل تحقق هستند، حال آن که در سطح بین المللی فقط تسهیلاتی برای پرداخت های کلان از طریق شبکه های تسویه حساب بین بانکی اندیشیده شده و پرداخت های خرد که حجم بالنسبه وسیعی را به خود اختصاص داده، جز در اتحادیه اروپا، مورد توجه مقررات بین المللی نبوده اند».



عنوان کتاب: اصول مجازی سازی و رایانش ابری

نویسندگان: محمدرضا احمدی، احسان

آریانیان و داود ملکی

ناشر: نیاز دانش

زمان نشر: ۱۳۹۲

تعداد صفحات: ۴۵۶ برگ

شمارگان: ۵۰۰ نسخه

شابک: ۹۷۸ - ۶۰۰ - ۶۴۸۱ - ۴۲ - ۵

قیمت: ۲۰۰۰۰ تومان

مقدمه

در پیشگفتار کتاب، آمده است: «فراهم نمودن امکانات لازم برای سرویس های مطلوب - رایانشی - سرمایه گذاری های فراوانی را طلب

سبب معتقدند که مناقشه در باره تغییرات آب و هوایی به شدت سیاسی و هراس آفرین شده است. سیمون جنکینز^۱ از مجله ساندی تایمز^۲ حس وحشت جمعی و ترس آفرینی ایجاد شده به واسطه مباحثی در باره گرمایش جهانی را این گونه توصیف می کند: دولت های کنونی به ما هشدار می دهند که باید از ایدز، تروریست ها و در حال حاضر، گرمایش زمین هراسان باشیم».

در فصل دوم تحت عنوان رویکرد جامع به فناوری اطلاعات سبب آمده است: «برای مقابله موثر و کارآمد با اثرات مخرب زیست محیطی، ما باید رویکرد جامعی را برای حل مشکلات با توجه به شش مسیر مکمل اتخاذ نمائیم: طراحی سبز، تولید سبز، استفاده سبز، دفع و بازیافت سبز، استانداردها و معیارهای سبز و استراتژی و سیاست های فناوری اطلاعات سبز».



عنوان کتاب: حقوق بانکداری الکترونیکی

مجری طرح: مصطفی السان

همکار اصلی: محسن علی دادی

ناشر: پژوهشکده پولی و بانکی بانک

مرکزی جمهوری اسلامی ایران

زمان نشر: تابستان ۱۳۹۲ (ویرایش دوم)

تعداد صفحات: ۵۰۴ برگ

شمارگان: ۱۰۰۰ نسخه

شابک: ۹۷۸ - ۶۰۰ - ۵۹۸۲ - ۶۱ - ۹

قیمت: ۲۰۰۰۰ تومان

مقدمه

در بخش کلیات کتاب، نوشته شده است: «بحث قراردادهائی که مشتریان با بانکها منعقد می کنند تا از خدمات بانکی اینترنتی بهره مند شوند، رابطه حقوقی آن ها با آی.اس.پی ها، فرآیند حقوقی استفاده از خدمات اینترنتی، چارچوب قانونی که بانکها می توانند در آن قالب به ارائه خدمات از طریق اینترنت بپردازند، بانک های مجازی، حقوق و تعهدات طرفین قرارداد و مسئولیت های مدنی، کیفری، بحث های مربوط به امنیت و همین طور قوانین حاکم بر روابط حقوقی، دادگاه صالح به رسیدگی، نوع دلایل اثباتی، و... همگی موضوع حقوق بانکداری اینترنتی را تشکیل می دهند».

محتوای کتاب

کتاب دارای مقدمه، خلاصه طرح، شرح کامل طرح، تعریف مسئله، فرضیات، هدف از تحقیق، دلایل ضرورت و توجیه انجام طرح، توضیح دقیق و مشروح شرح روش پژوهش و تکنیک های اجرایی و شرح خدمات تفصیلی پروژه و هشت فصل مطلب با عناوین زیر است:

1- Simon Jenkins

2- Sunday Times

از کارکردها و ویژگی‌های ابر جهت تقویت سازمان، آزمون فرآیند و نتایج مهاجرت و تکرار و بهینه‌سازی».



عنوان کتاب: طراحی و پیاده‌سازی

زبان‌های برنامه‌نویسی

نویسنده: علیرضا خلیلیان

ناشر: پوران پژوهش

زمان نشر: (چاپ چهارم) ۱۳۹۵

تعداد صفحات: ۶۳۰ برگ

شمارگان: ۵۰۰ نسخه

شابک: ۹-۳۲۶-۱۸۴-۹۶۴-۹۷۸

قیمت: ۲۹۰۰۰ تومان

مقدمه

در پیشگفتار کتاب، نویسنده نوشته است: «از زمان نگارش مرجع اصلی این درس در سال ۲۰۰۱ تا کنون، کتاب‌ها و مقاله‌های بسیاری پیرامون طراحی و پیاده‌سازی زبان‌ها منتشر شده است. زبان‌های متعدد جدیدی طراحی شده‌اند و ویژگی‌های جدیدی به زبان‌های موجود اضافه شده‌اند... چاپ چهارم در حقیقت ویراست جدیدی از کتاب هم هست... چاپ چهارم سعی کرده تا ذائقه خوانندگان مختلف خود را بیشتر مورد توجه قرار داده و آن چنان باشد که هم یک متن کامل درسی (صد البته پیشنهادی) برای درس طراحی زبان‌ها در دوره کارشناسی باشد و هم مورد استفاده متقاضیان ورود به دوره تحصیلات تکمیلی مهندسی نرم‌افزار قرار گیرد».

محتوای کتاب

کتاب دارای پیشگفتار انتشارات و پیشگفتار مولف، فهرست مطالب و یازده فصل مطلب و فهرست مراجع است. عناوین فصل‌های کتاب به شرح زیر است:

- فصل ۱: اصول طراحی زبان‌ها.
- فصل ۲: تکامل زبان‌های برنامه‌سازی مهم.
- فصل ۳: پردازش زبان‌های برنامه‌سازی.
- فصل ۴: اصول ترجمه و مدل‌سازی خواص زبان.
- فصل ۵: نوع داده.
- فصل ۶: صفات کنترل داده‌ها.
- فصل ۷: انواع داده انتزاعی.
- فصل ۸: برنامه‌نویسی شیء‌گرا.
- فصل ۹: کنترل ترتیب اجرا.
- فصل ۱۰: کنترل زیربرنامه.
- فصل ۱۱: پیاده‌سازی زیربرنامه.

می‌کند. به کارگیری منابع سخت‌افزاری و نرم‌افزاری موجود، متناسب با نیازهای گسترده موجود بسیار دشوار بوده و استفاده از روش‌های معمول، پیچیدگی و اتلاف سرمایه گذاری فراوانی را ایجاب خواهد نمود... مجازی‌سازی یکی از ایده‌های موثر و مفید است که می‌تواند به عنوان یک تکنولوژی جدید مطرح گردد. این تکنولوژی امکان تجمع و یکپارچگی منابع سخت‌افزاری نرم‌افزاری مختلف را فراهم نموده و آن‌ها را برای اجرای وظایف گسترده آماده می‌سازد. به طوری که همه کاربران احساس دسترسی به محیط‌های اختصاصی خود را دارند... از طرف دیگر رایانش ابری یک مدل رایانشی بر پایه شبکه‌های بزرگ کامپیوتری مانند اینترنت است که الگویی تازه برای عرضه، به کارگیری و ارائه سرویس‌های فناوری اطلاعات شامل سخت‌افزار، نرم‌افزار، محتوا و سایر منابع اشتراکی رایانشی را ارائه می‌کند. رایانش ابری راهکارهایی برای ارائه خدمات فناوری اطلاعات مشابه با سرویس‌های همگانی مانند آب، برق، تلفن و غیره پیشنهاد می‌کند».

محتوای کتاب

کتاب دارای پیشگفتار و دوازده فصل در دو بخش، مراجع، واژه‌نامه، کوتاه نوشته‌ها، فهرست اشکال، فهرست جداول است. عناوین بخش‌ها و فصل‌های کتاب به شرح زیر است:

- بخش یک: مجازی‌سازی.
- فصل ۱: مفاهیم و تعاریف پایه در مجازی‌سازی.
- فصل ۲: تکنولوژی‌های مختلف مجازی‌سازی.
- فصل ۳: مجازی‌سازی حوزه کارساز.
- فصل ۴: مجازی‌سازی حوزه ذخیره‌سازی و شبکه.
- بخش دوم: رایانش ابری.
- فصل ۵: مفاهیم و تعاریف پایه در رایانش ابری.
- فصل ۶: فناوری و ابزارهای رایانش ابری.
- فصل ۷: مدل مفهومی رایانش ابری.
- فصل ۸: سرویس‌های رایانش ابری و بررسی ویژگی‌های نرم‌افزاری برای ایجاد آن‌ها.
- فصل ۹: فناوری‌های متداول برای ایجاد زیرساخت رایانش ابری.
- فصل ۱۰: امنیت در رایانش ابری.
- فصل ۱۱: مسائل حقوقی در رایانش ابری.
- فصل ۱۲: استراتژی‌های ایجاد رایانش ابری و مهاجرت سرویس‌ها به بستر ابر.

در فصل دوازدهم کتاب به عنوان سناریوی دوم مهاجرت به فضای رایانش ابری گام‌های زیر تشریح شده است: «...قدم‌های لازم برای به کارگیری این مدل در یک فرآیند هفت گامی تکرارپذیر شامل قدم‌های زیر است: برآورد عملیات مهاجرت به ابر، جداسازی وابستگی‌ها، نگاشت بین محیط قدیمی و محیط ابر، معماری مجدد و پیاده‌سازی عملیات از دست رفته، استفاده

چگونه دنیا را تغییر دهیم؟

مدیریت تغییرات ۳،۰

(قسمت اول)

نوشته: یورگن آپلو

ترجمه: علی حاجی‌زاده مقدم

پست الکترونیکی: ali.moghadam@gmail.com

یادداشت سردبیر

کتاب چگونه دنیا را تغییر دهیم؟ توسط آقای مهندس علی حاجی‌زاده مقدم و با دریافت مجوز از نویسنده کتاب، به فارسی برگردانده شده و نسخه الکترونیکی آن در نشانی اینترنتی www.adanic.ir/download در دسترس همگان قرار داده شده است. مترجم، این کتاب را برای چاپ در اختیار انجمن انفورماتیک ایران گذاشته است که ضمن سپاسگزاری از ایشان، از این شماره به صورت یک مقاله دنباله‌دار تقدیم خوانندگان گزارش کامپیوتر می‌گردد.

* * *

مقدمه نویسنده

سوالاتی که خیلی‌ها شاید داشته باشند این است: چطور با مشکلات سازمانی که در آن هستم سر و کله بزنم؟ من از کارم خوشم می‌آید، ولی کارهایی را که مدیریت سازمان انجام می‌دهد دوست ندارم. چکارش کنم؟

جواب ساده است. شما سه گزینه دارید:

۱- نادیده بگیریدشان. تغییر دادن سازمان‌ها خیلی کار سختی است. اگر توان و طاقت یادگرفتن این‌که چطور باید یک تغییر دهنده بود را ندارید، شکایت کردن درباره مشکلات را متوقف

کنید. سازمان را همان طور که هست بپذیرید و از نقاط مثبت کارتان لذت ببرید. در این صورت می‌توانید خواندن این متن را همین جا متوقف کنید.

۲- از کارتان استعفا دهید. تنها دلیل این‌که سازمان‌های بد وجود دارند این است که مردم به کار در آن‌ها ادامه می‌دهند. با یافتن یک جای بهتر، شما به دنیا هم لطف می‌کنید. به سازمان‌های ناجور هم با کار نکردن برایشان کمک کنید که بهتر شوند.

۳- درباره مدیریت تغییرات بیاموزید. بیشتر مردم در اثرگذاری بر دیگران و تغییر دادن سازمان‌ها ضعف دارند. ولی اگر شما در این باره جدی هستید، می‌توانید بیاموزید که چطور به شکل موثرتری یک تغییر دهنده شوید.

آن طور که دوستم اولاف لویتر گفته، «یا قبولش کن، یا ترکش کن، و یا تغییرش بده...»

این کتاب را برای کسانی نوشته‌ام که گزینه سوم را انتخاب می‌کنند.

یورگن آپلو

روتردام، مارچ ۲۰۱۲

jurgen@noop.nl

از شکست تا شیدایی

من ایده روشنی ندارم که چطور باید مردم را عوض کنم، ولی برای روزی که راهش را یافتم، لیستی طولانی از نامزدهای این کار دارم.
دیوید سداریس، نویسنده آمریکایی (۱۹۵۶)

ادواردز دمی‌نگ ده‌ها سال قبل نوشت که روش‌های متداول دادن پاداش، برای کسب و کارها مفید نیستند [Deming 1986]. ولی بیشتر مدیران در دنیا هنوز از آن استفاده می‌کنند. پیتر دراگر مدت‌ها پیش گفت که کارکنان دانش ورز^۱ نباید زیر دست مدیران باشند [Drucker 1974]. ولی مدیرها هنوز طوری رفتار می‌کنند که انگار آن‌ها مافوق دیگران هستند. تحقیقات بارها و بارها به ما نشان داده که روش‌های ارزیابی عملکرد^۲ درست کار نمی‌کنند [Bobinski 2010]. ولی بسیاری از مدیران هنوز بر استفاده از آن‌ها به عنوان تکنیک اصلی سنجش اصرار دارند.

چرا؟

چرا مدیریت این قدر کند تغییر می‌کند (یا اصلاً تغییر نمی‌کند)؟
بیانیه چابک^۳ در ده سال گذشته اثر بسیار عمیقی بر توسعه نرم افزار در همه جای دنیا گذاشته است. البته هنوز کار به سرانجام نرسیده است، ولی پیشرفت‌هایی داشته ایم. منتها به نظر می‌رسد جایگزین شدن روش‌های مدیریتی سنتی با بُن‌انگاره (پارادایم) جدید به اندازه ذوب شدن اقیانوس منجمد شمالی طول می‌کشد!

ایده‌ها برای بهبود سازمان‌ها ده‌ها سال است که مطرح اند ولی هنوز تغییرات در مدیریت کسب و کارها بسیار آهسته انجام می‌شوند. آیا ما می‌توانیم سرعت تغییر روش اداره سازمان‌ها را بیشتر کنیم؟ آیا می‌توانیم به حرکت سراسری برای تغییر، انرژی بیشتری بدهیم؟

همراه با بسیاری دیگر از نظریه‌پردازها و فعالان مدیریت، من هم به دنبال روش‌هایی برای انرژی دادن به حرکت تحول سراسری در سازمان‌ها هستم. این به معنی تغییر کلان روش‌های مدیریتی در مقیاس جهانی و با ابعاد حماسی است. بلند پروازانه است؟ بله. دشوار است؟ مسلماً. امکان پذیر است؟ نمی‌دانم. ولی حداقل این به من انگیزه تلاش می‌دهد. در این کتاب من رویکردی به مدیریت تغییرات را ارائه می‌کنم که ثابت شده در مقیاس کوچک تر کار می‌کند. ممکن است در مقیاس جهانی هم کار کند. می‌تواند به شما در تلاشتان برای اصلاح سازمانتان کمک کند. ولی بدون شک این روش به من در تغییر سازمانم کمک کرده است. ولی قبل از شروع، کمی به شما درباره دفعات بسیاری که در تغییر محیطم شکست خوردم خواهم گفت.

۱۵ سال شکست

من دوست دارم درباره این حقیقت که برای ۱۵ سال شکست خورده‌ام اعتراف کنم. در سال ۱۹۹۲ من تلاش کردم روزنامه‌ای بین‌المللی درباره

۱- Knowledge Workers: کارکنان فنی و حرفه‌ای که کار اصلی آن‌ها توسعه و استفاده از دانش است.

۲- performance appraisals

۳- <http://agilemanifesto.org/>

موسیقی ایجاد کنم، و شکست خوردم. در ۱۹۹۴ کوشیدم یک شرکت توسعه بازی‌های رایانه‌ای راه بیندازم و شکست خوردم. سه سال را صرف نوشتن یک برنامه کتابداری کردم، ولی مشتریان اندکی یافتم. در سال ۱۹۹۷ به همراه یک دوست دیگر، یک شرکت نرم‌افزاری تاسیس کردیم که برای مدتی بسیار موفق بود. ولی ناگهان این شرکت به همراه شرکت مادرش از هم فروپاشید (همانند ده‌ها رابطه عاطفی که من در آن دهه داشتم).

اولین نشانه‌های تغییر در ۱۹۹۹ از راه رسید، وقتی که من یک شرکت نوپای اینترنتی راه انداختم. برنامه کسب و کار من به قدری جذاب بود که یک جایزه ملی برایش بردم، و یک میلیون یورو سرمایه جذب کردم. من و تیمم موفق شدیم تا آخرین یورو را خرج کنیم. ولی در عین حال ما نتوانستیم هیچ مشتری پیدا کنیم و این کسب و کار هم فروپاشید. من چیزی از شکست‌های دیگرم در فاصله ۲۰۰۰ تا ۲۰۰۸ برایتان نمی‌گویم که شامل یک رمان، یک کارتون، یک وب‌نوشت، یک کتاب دیگر، و چندین پروژه نرم‌افزاری بکر، ولی شکست خورده دیگر بودند. بارها و بارها تلاش کردم اثری روی دنیای اطرافم بگذارم و هر بار شکست خوردم. آیا تسلیم شدم؟ البته که نه!

از سال ۲۰۰۸، به دلایلی غیرقابل توضیح، شرایط شروع به تغییر کرد. وب‌نوشت من Noop.nl تبدیل شد به یکی از موفق‌ترین‌ها، و رتبه سوم را بین ده وب‌نوشت برتر چابک در دنیا کسب کرد [Saddington 2010]. کتاب من مدیریت ۳،۰ هم موفقیت دیگری بود که به دعوتنامه‌های فراوان برای سخنرانی در سمینارها و کنفرانس‌ها منجر شد. دوره آموزشی من هم به نظر موفقیت آمیز می‌رسید و در نقاط مختلف دنیا ارزیابی‌ها درباره آن بسیار مثبت بود. رابطه شخصی آخرم هم آنقدر خوب بود که به ازدواج منتهی شد. ایده من برای تاسیس شبکه چابک اروپا منجر به کنفرانس اروپایی ALE 2011 در برلین با بیش از ۲۰۰ شرکت کننده شد.

گاهی مردم از من می‌پرسند: چه چیزی تغییر کرد؟ یا: چطور این کار را کردی؟

راستش من خودم هم در این باره شگفت زده‌ام...

شاید من نیرویی را که در کائنات صرف شکست من می‌شد خسته کرده‌ام. شاید کسی آن بالا سرش را تکان داده و پیش خودش فکر کرده: این مقدار شکست برایت بس است. اگر همینطور ادامه بدهی هیچ چیز برای دولت‌های اروپایی باقی نمی‌ماند.

تنها چیزی که می‌توانم با اطمینان بگویم این است که من الان زمان بیشتری را صرف یاد گرفتن این می‌کنم که چطور می‌توانم یک نویسنده بهتر، یک سخنران بهتر، و یک مدرس بهتر باشم. و همینطور یک تغییر دهنده بهتر!

دلیلش هر چه باشد، بعد از این همه تلاش ناموفق برای تحقق ایده‌هایم، من یکی دو نکته مهم درباره این‌که چطور عاملی برای تغییرات نسبتاً موفق ولی قطعاً بسیار ماندگار باشم یاد گرفتم. در این کتاب می‌خواهم

فوت و فن سر و کار داشتن با احتمالات رفتاری در سیستم‌های پیچیده، این است که ببینیم چه ساختارهایی شامل چه رفتارهای پنهانی هستند و چه شرایطی این رفتارها را آزاد می‌کنند اگر هم امکان پذیر باشد، ترتیب دادن ساختارها و شرایطی برای کاستن از احتمال بروز رفتارهای مخرب و تشویق احتمال رفتارهای مفید. آدمهای باهوش می‌گویند: «شما نمی‌توانید افراد را وادار کنید رفتار متفاوتی داشته باشند.» درست است، ولی همان طور که ما نمی‌توانیم دیگران را «وادار کنیم بخندند» یا «وادار کنیم شاد باشند»، حداقل می‌توانیم برایش تلاش کنیم. این همه چیز است که مدیریت تغییرات درباره‌اش حرف می‌زند.

روش موهیتو

من بارها در وب‌نوشت و کتاب‌هایم گفته‌ام که دیگر نوشیدنی‌های الکلی نمی‌نوشم. فقط یک استثنا دارم: موهیتو. همیشه امیدوارم یک روز با مزه تازه یک موهیتوی خوب در اتاق هتل شگفت زده شوم. متأسفانه هیچ وقت این قدر خوش شانس نبوده‌ام. اما چون آدم خوشبینی هستم، امیدم را از دست نمی‌دهم.

چیزی که درباره موهیتو جالب است این است که محتویات آن هریک به تنهایی چیزهای نامطبوعی هستند. من فکر می‌کنم ایده‌ها هم همین طور هستند. وقتی شما ایده‌هایی از منابع مختلف را مخلوط می‌کنید، ایده جدیدی پدیدار می‌شود که ایده‌های اولیه را هم یکپارچه می‌کند و هم بهبود می‌دهد.

من این روش را روش موهیتو می‌نامم.

دوستانم به من می‌گویند موهیتوهای خوبی درست می‌کنم. من در به کار بردن روش موهیتو هم مهارت دارم. ایده‌ها را از آدم‌های دیگر می‌گیرم، آن‌ها را با هم قاطی می‌کنم، و به شکلی که (به نظر خودم) مفیدتر از تک تک ایده‌هاست به کار می‌برم. دلیلی که این روش جواب می‌دهد را در کتابم گفته‌ام:

تحقیقات اخیر نشان داده که کپی کردن ایده‌ها موفق‌ترین راهبرد برای بهبود است. این نشان می‌دهد تیم‌ها باید بیشتر زمان یادگیری خود را صرف کپی کردن ایده‌ها از منابع دیگر کنند. فقط بخش کوچکی از زمان را باید برای اختراع ایده‌های جدید صرف کنند.

روش موهیتو برای هرکسی که می‌خواهد بدون تحمل مشقت اختراع همه چیز، نوآوری کند، جواب می‌دهد. من از این روش برای ساختن دیوار خوشحالی (ترکیب دیوار بازخورد و شاخص خوشحالی)، پوکر واگذاری مسئولیت^۵ (ترکیب مدیر خدمت‌گزار، حوزه‌های اصلی تصمیم^۶ و پوکر برنامه‌ریزی)، و انگیزه‌های متحرک (برگرفته از ۱۶ میل اصلی [Reiss 2002] و نظریه خودمختاری^۷ [Deci, Ryan 2002]) استفاده کرده‌ام.

درس‌هایی را که گرفتم با شما در میان بگذارم. امیدوارم این به شما کمک کند که سریعتر از من بیاموزید.

چطور دیگران را تغییر دهیم؟

سخت‌ترین بخش در فرآیند بهبود مداوم سازمان‌ها، تغییر رفتار اشخاص دیگر است (حداقل این چیزی است که دیگران مرتب به من می‌گویند). وقتی اشخاص می‌خواهند دانش ورزش‌های حرفه‌ای‌تری شوند، یا محیط کار قابل تحمل‌تری بسازند، یا کاری را انجام دهند که از آن واقعا لذت ببرند، سازمان‌های آن‌ها (سیستم اطراف آن‌ها) همیشه همکاری نمی‌کنند. چطور می‌توانید مدیریت خود را «وادار کنید» چابک‌تر باشد؟ چطور می‌توانید اعضای تیم را قانع کنید که خودشان را بهتر آموزش دهند؟ چطور می‌توانید باعث شوید همکارانتان به جای رقابت با هم همکاری کنند؟ چطور می‌توانید مشتریان را به سمت قراردادهای منصفانه و همکاری سوق دهید؟

من فهمیده‌ام که همه این چالش‌ها به یک سوال واحد ختم می‌شوند: چطور می‌توانم رفتار آدم‌های دیگر را تغییر دهم؟ آیا رفتار دیگران چیزی است که شما واقعا می‌خواهید عوض کنید؟ نمی‌خواهید باعث شوید اول آن‌ها مسئله را بفهمند؟ نمی‌خواهید ارزش‌ها و اعتقاداتشان را عوض کنید؟ اگر در تغییر اندیشه‌ها موفق شوید، آن‌ها خودشان رفتارشان را عوض خواهند کرد و البته این ممکن است به رفتارهایی منتهی شود که شما انتظار نداشتید، ولی حتی ممکن است از آن چه شما امید داشتید هم بهتر باشد.

به نظر می‌رسد ما داریم درباره مدیریت تغییرات صحبت می‌کنیم. ما همه می‌خواهیم بدانیم که چطور می‌شود یک سیستم پیچیده اجتماعی را تغییر داد؟ یا چطور می‌توان دنیا را تغییر داد؟

خوشبختانه کتاب‌های زیادی درباره این موضوع وجود دارد. سویچ [Heath 2010]، تغییر بدون ترس [Manns, Rising, 2005]، رهبری تغییر [Kotter, 1996]، اثرگذار [Patterson 2008]، و بسیاری کتاب‌های دیگر، که صدها صفحه راهنمایی‌های مفید برای تغییر دهنده‌ها دارند. با وجود این که تعدادی از این کتاب‌ها خیلی خوب هستند، اغلب تصویری بسیار ساده شده از تغییر در سیستم‌های اجتماعی ارائه می‌دهند.

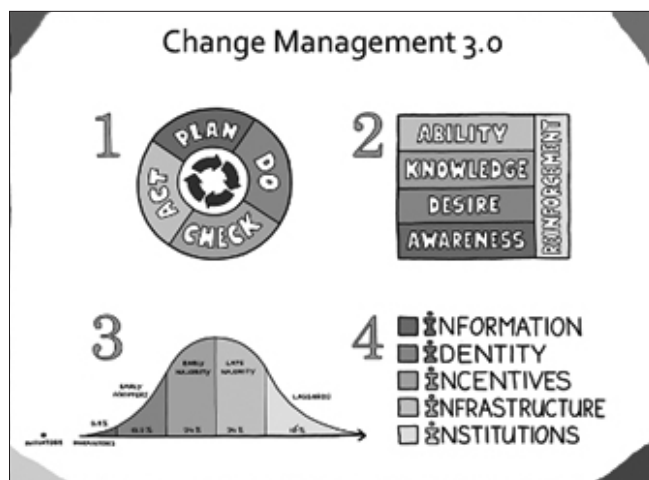
تفکر پیچیدگی

مدل‌های ساده که توسط داستان‌های الهام بخش کامل شده‌اند، برای شروع مبانی مدیریت تغییرات خوب هستند. اما دنیای واقعی بسیار پیچیده‌تر از چیزی است که بیشتر مدل‌ها به شما می‌گویند. ما به رویکردی کامل‌تر برای تغییرات سازمانی نیاز داریم. خیلی سخت است که پیش‌بینی کنید یک سیستم اجتماعی پیچیده چطور رفتار خواهد کرد. ما نیاز داریم بفهمیم چطور می‌توانیم با سقلمه زدن به یک جای سازمان، کل آن را متاثر کنیم. بعد می‌بینیم چطور پاسخ می‌دهد. به عنوان یک تغییر دهنده، ما تلاش می‌کنیم به اشخاص، تیم‌ها و سازمان‌ها سیخونکی بزنیم تا آن‌ها خود را از نو سازماندهی کنند.

5-Delegation Poker

6-Key Decision Areas

7-Self-determination theory



شکل ۱: چهار جنبه مدیریت تغییرات ۳,۰

را به کار ببرید (شکل ۱). من به آن آبرمدل می گویم، چون شامل چند مدل شناخته شده کوچک تر برای تغییر است که هر کدام به یکی از جنبه هایی که من اشاره کردم می پردازند. این روش موهیتو است که در مدیریت تغییرات به کار گرفته شده است. با سیستم برقص - با استفاده از مدل PDCA (برنامه ریزی/اجرا/بررسی/کنش)

به آدمها اهمیت بده - با استفاده از مدل ADAKR (آگاهی/میل/دانش/توانایی/تقویت)

شبکه را برانگیز - با استفاده از مدل منحنی سازگاری^۹
محیط را تغییر بده - با استفاده از مدل (شفافیت اطلاعات/شناسه/مشوق ها/شالوده/تشکیلات)

من هر کدام از این چهار مدل کوچک تر را در چهار فصل بعدی توضیح داده ام. وقتی شما این آبرمدل را درک کنید، برای تلاش در تربیت افرادی چابک تر، آموخته تر، با همکاری بیشتر، یا (اگر از یک چالش حساسی خوشتان می آید) مشتاق تر به اجرای یک سرود دسته جمعی آماده خواهید بود.

با سیستم برقص

مهم است که هیچ وقت هدف به شکل فعالیت یا روشها تعریف نشود. هدف باید همیشه مستقیماً به این که چطور زندگی برای همه بهتر می شود ربط داشته باشد. هدف سیستم باید برای همه داخل سیستم روشن باشد. هدف باید شامل برنامه هایی برای آینده باشد. هدف یک قضاوت ارزشی است.

ادواردز دمنینگ، آماردان آمریکایی (۱۹۰۰-۱۹۹۳)

در ژانویه ۲۰۱۲، گروهی از رهبران فکری قرن ۲۱، باغبانهای ایده، مربیان مدیریت، مشاوران، و حقوقدانها، در شهر استوس در سوئیس دور هم جمع شدند. هدف چنین بود: «بحث درباره چگونگی سرعت بخشیدن به تغییرات مدیریتی و سازمانی». این رخداد توسط استیو دنینگ، فرانز روسلی، پیترا استیونز، و این ارادتمند شما هماهنگ شده

9- Adaption Curve

حالا می خواهیم روش موهیتو را روی مدیریت تغییرات اعمال کنیم. بگذارید ببینیم می توانم به شما در شکستن مقاومت در برابر تغییر و مخلوط کردن چیزها کمک کنم یا نه.

مدیریت تغییرات ۳,۰

یک سازمان، ماشینی نیست که توسط صاحب آن رانده می شود بلکه یک شبکه اجتماعی است. اشخاص در همه سطوح ساختار سازمانی با هم تعامل می کنند.

وقتی می خواهید رفتارهای چیزی به پیچیدگی یک سیستم اجتماعی را تغییر دهید، باید چهار جنبه مدیریت تغییرات را درک کنید.

اول: با سیستم برقص

یک شبکه اجتماعی، پیچیده و سازگاری پذیر^۸ است [Mitchell 2009]. شبکه با کارهای شما سازگار می شود و شما هم باید با شبکه سازگار شوید. این کار مانند رقصیدن با سیستم است. کسی که درباره پیچیدگی فکر کرده باشد می فهمد که تغییرات به ندرت از یک مسیر سر راست پیروی می کنند و معمولاً مسیرشان دشوار و سنگلاخ است.

دوم: به افراد اهمیت بده

یعنی درک این که آدمها سخت ترین بخش یک سیستم اجتماعی هستند و با هم فرق دارند. هیچ روش یکپارچه ای که به همه بخورد وجود ندارد. اغلب اوقات هم این که از آدمها درخواست کنیم تغییر کنند اثری ندارد. تنوع چیزی است که باعث می شود سیستمهای پیچیده کار کنند و به همین دلیل وقتی با آدمها سر و کار داریم به روشهای متنوع نیاز داریم.

سوم: شبکه را برانگیز

فهمیدن این که چطور رفتارها در یک سیستم پیچیده منتشر می شوند. این در یک شبکه اجتماعی به آدمها و رفتارهای متقابلشان مربوط می شود. رفتارها مثل ویروس منتشر می شوند، و برانگیختن شبکه اجتماعی می تواند بر مقاومت در برابر تغییر غلبه کند و کل سازمان را تغییر دهد.

چهارم: محیط را تغییر بده

افراد معمولاً خودشان را در چارچوب یک محیط سازماندهی می کنند. محیط مشخص می کند که سیستم چطور می تواند خودش را سازماندهی کند و شما ممکن است بتوانید محیط را انگولگ کنید. از آنجا که رفتار افراد به محیط آنها وابسته است، با تغییر محیط افراد را تغییر می دهید.

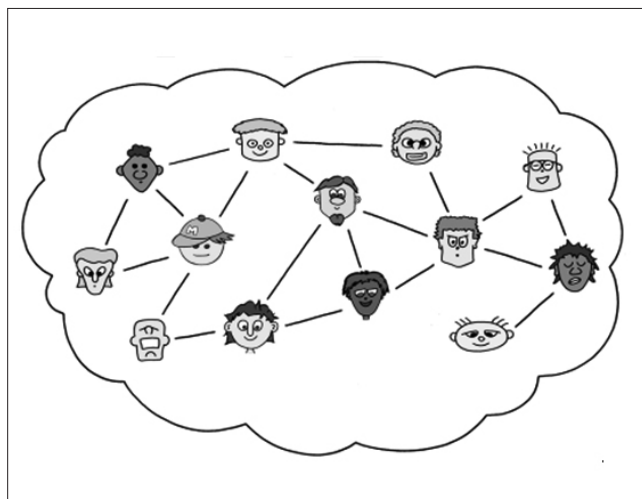
آبرمدل مدیریت تغییرات ۳,۰

اگر شما این جنبه های مدیریت تغییرات را درک می کنید، در این صورت به شما تبریک می گویم، شما می توانید آبرمدل مدیریت تغییرات ۳,۰

8-Adaptive



شکل ۳: مدل PDCA - برنامه ریزی، انجام، بررسی، کنش



شکل ۴: یک سازمان به عنوان یک سیستم سازگاری پذیر پیچیده

آن‌هایی که خوب هستند و چه آن‌هایی که بد هستند. وقتی مردم رفتار شما را نقد می‌کنند، معنی اش این است که به موضوع اهمیت می‌دهند. اگر خوانندگان و پیگیری کنندگان سکوت می‌کردند نگران کننده بود. افزون بر این، همان طور که یک نفر گفته بود، بحث‌های داغ بازتاب پیچیدگی مسئله‌ای بودند که ما به آن می‌پرداختیم. اگر همه به سادگی موافق بودند، از اول اصلاً مشکلی وجود نداشت!

در این فصل، به عمق موضوع بهبود (یا تغییر) مداوم شیرجه می‌زنیم.

بازبینی و سازگاری (و پیش بینی و کاوش)

یک سازمان، یک سیستم پیچیده سازگاری پذیر است (شکل ۲). سیستم با رفتار شما سازگاری پیدا می‌کند و شما هم باید به طور مداوم با پاسخ سیستم سازگار شوید. این در واژه‌های بازبینی و سازگاری، که در جامعه چابک زیاد استفاده می‌شوند، منعکس شده است.

ولی... سازگاری پذیری کافی نیست. اگر سازگاری پذیری تنها کاری باشد که شما انجام می‌دهید، معنی اش این خواهد بود که سیستم همیشه در پیش است (و شما در پس آن). کسانی که هرگز سیستم را هدایت نمی‌کنند، تنها نظاره گر مسیر حرکتش خواهند بود! به عنوان یک تغییر دهنده، باید پیش بینی و کاوش را به سازگاری پذیری بیفزائید. به عبارت دیگر...

ما نمی‌توانیم سیستم‌ها را کنترل کنیم یا آن‌ها را درک کنیم. بلکه می‌توانیم با آن‌ها برقصیم!

مدل PDCA

این یک مدل جدید نیست. در واقع، برخی از اندیشمندان قدیمی‌تر سیستم‌ها یعنی ادواردز دمنینگ و والتر شوارتز این مدل را ده‌ها سال قبل بیان کرده‌اند.

توسط شوارتز در اوایل دهه ۱۹۲۰ توسعه داده شد و توسط دمنینگ در دهه ۱۹۵۰ مشهور شد.

PDCA مخفف کلمات برنامه ریزی، اجرا، بررسی، کنش، یک فرآیند

بود. این طاق‌فراست‌ترین و البته مفیدترین تجربه یادگیری بود که من بعد از مدت‌ها داشتم.

سواً پشتیبانی‌های مشتاقانه و پیام‌های دلگرم کننده، شرکت کنندگان در استوس با مقداری واکنش منفی، هم در طول گردهمایی و هم پس از آن، روبرو شدند. ما متهم شدیم به «باز و شفاف نبودن».

ما چه اشتباهی انجام داده بودیم؟

اگر بخواهیم رو راست باشیم، باید بگویم برگزار کنندگان (از جمله خودم) به خاطر اعلام عمومی برگزاری یک رویداد خصوصی، و درخواست نظر از خوانندگان و پیگیری کنندگان گناهکار بودیم. چنین اعلان (و درخواست نظری)، در رویدادهای خصوصی قبلی اتفاق نیفتاده بود؛ مثل همان رویدادهایی که به بیانیه چابک و اعلامیه وابستگی متقابل^{۱۰} ختم شده بودند. متأسفانه این برخورد باز قبل از رویداد با نیاز برخی از دعوت شدگان به رعایت حریم خصوصی‌شان در طول برگزاری رویداد در تضاد بود. به عنوان یک گروه، ما تصمیم گرفتیم به این نیاز به رعایت حریم خصوصی احترام بگذاریم که به سکوت (نسبی) ما در شبکه‌های اجتماعی طی زمان جلسات خصوصی منجر شد. به هر حال، با وجود این که رویدادهای قبلی در خلال جلساتشان چیزی را منتشر نمی‌کردند، برخی افراد سکوت ناگهانی ما را ناراحت کننده یافتند. و آن‌ها این ناراحتی را بلند و رسا بیان کردند. هیچکدام از برگزار کنندگان این را پیش بینی نمی‌کردند.

در حقیقت، این بازخورد قابل درک و بدیهی بود. در تفکر پیچیدگی، ما به آن چسبندگی گذشته نگر^{۱۱} می‌گوییم. ما آن را پیش بینی نمی‌کردیم، ولی پس از وقوع قابل درک بود. متأسفانه، بدون تجربه مستقیم برگزاری چنین رویدادی، مدیریت این ناشناخته‌ها غیر ممکن است. ما فقط می‌توانیم بیاموزیم و در چرخه تکرار، بازبینی و سازگاری^{۱۲} بهبودش دهیم. برای یک تغییر دهنده مهم است که به همه بحث‌ها اهمیت بدهد، چه

10- Declaration of Interdependence, <http://pmdoi.org/>

11- Retrospective Coherence

12- Iterate, Inspect and Adapt

بهبود تکرار شونده^{۱۳} چهار مرحله ای است. هریک از ابتکارات شما برای مدیریت تغییرات، شامل مجموعه ای از دخالت‌ها در سیستم (یا محیط آن) است، و شما اثر این دخالت‌ها را بر سیستم (یا بر خودتان) بررسی می‌کنید. وقتی کارهای شما همان اثری را داشت که می‌خواستید، قدم بعدی را بر می‌دارید. وقتی کارهایتان جواب نداد، کار دیگری را می‌آزمایید. این چرخه ای است بی پایان از مشخص کردن هدف‌ها (برنامه ریزی)، پیاده سازی ایده‌ها (انجام)، اندازه گرفتن بازخورد (بررسی)، و کاویدن نتایج (کنش) (شکل ۳).

رقصیدن با سیستم درباره رهبری و پیروی کردن است. چرخه ای در جریان است از برنامه ریزی برای حرکت در یک جهت، برداشتن یک قدم، سنجیدن پاسخ، و ارزیابی موفقیت. ترکیبی است از پیش بینی، سازگاری، و اکتشاف.

مفهوم PDCA بر روش تجربی علمی بنا شده است: فرضیه، آزمون، و ارزیابی. مدل‌های جایگزین زیادی وجود دارند، مانند ارزیابی-تحلیل-سنجه پذیری^{۱۴} -بهبود [Pulford 1996]، تعریف-اندازه‌گیری -بهبود-کنترل-تحلیل^{۱۵}، و ساخت-اندازه‌گیری-یادگیری [Ries 2011]، ولی من ترجیح می‌دهم به همان مدل قدیمی برنامه‌ریزی-انجام-بازبینی-کنش بچسبم، چون ساده و برای مقصود من به اندازه کافی موثر است.

بخصوص برای رقصندگان بی تجربه، نتایج ناگزیر رفتن در یک مسیر اشتباه، لگد کردن انگشتان دیگران، یا تداخل با رقص دیگران، چالشی‌ترین جنبه مدیریت تغییرات است. فقط یک چاره برای این مشکل وجود دارد: رقصندهٔ بهتری باشید.

برنامه‌ریزی

من فکر نمی‌کنم که مردم تغییر را دوست ندارند. ظهور اینترنت، آی‌پد، و لیدی گاگا نشان می‌دهد که مردم تغییر را دوست دارند. اما من دوست دارم اعتقاد داشته باشم که مردم پیشنهادها را برای تغییر را دوست ندارند. من کاملاً مطمئن هستم که آن‌ها به شکل فعالی در برابر تغییرات مورد نظر شما مقاومت می‌کنند، چرا که قانع نشده‌اند که باید همراه شما تغییر کنند. بگذارید ببینیم چطور می‌شود این مشکل را حل کرد. هدف شما چیست؟

پیش از همه، برای واداشتن مردم به حرکت نیاز به یک هدف وجود دارد. آن‌ها چیزی لازم دارند که به خاطرش تغییر کنند. خیلی حیاتی است که روشن باشد قرار است چی به دست بیاورید. آیا می‌توانید یک تصویر از جهانی بهتر بکشید؟ یا یک سازمان بهتر؟ آیا می‌توانید چشم‌انداز و توصیفی بیان کنید که مردم بخواهند آن را باور کنند؟ و آیا می‌توانید از قصه‌ها، استعاره‌ها، و تصویرها برای رساندن بهتر پیام خود استفاده کنید؟ وقتی یک چشم‌انداز واقعی داریم (برعکس همه «اسناد چشم‌انداز» متداول)، مردم یاد می‌گیرند و بهتر می‌شوند، نه به خاطر این که به آن‌ها

گفته شده، بلکه چون می‌خواهند.

در فوریه ۲۰۱۱ من از فعالین چابک و ناب^{۱۶} در اروپا خواستم که همکاری‌های فرامرزی را بهبود دهند، چون به نظرم می‌رسید ارتباط کمی بین گروه‌ها در کشورهای مختلف وجود دارد. من چشم‌اندازی از گروه‌های اروپایی که با هم کار می‌کنند ساختم. در چند پست وب‌نوشت، توضیح دادم که با ترسیم تصویر دنیایی بهتر، چه چیزی به دست می‌آوریم [Appelo 2011e]. و جواب داد! چند ماه بعد ۴۰ نفر از همه جای اروپا دور هم جمع شدند و امیدها و رویاهایشان را برای اروپای چابک و ناب به شکل یک مدل لگوی فیزیکی درآوردند (شکل ۴)، با توضیحی در یوتیوب^{۱۷}. این تصویرسازی فیزیکی کمک کرد که چشم‌انداز به شکلی ملموس تر و عینی تر به اشتراک گذاشته شود.

کجا کار خوب پیش می‌رود؟

وقتی مقصد را فهمیدید، بهترین راه برای شروع این است که مثالی پیدا کنید که در جایی کارها خوب پیش می‌رود، بعد رفتارهای خوب را کپی کنید. تلاش کنید وضعیتی را بیابید که مثالی درخشان از چیزهایی باشد که شما می‌خواهید، یا به عبارت دیگر، یک نقطه روشن از رفتاری خوب پیدا کنید.

در بسیاری از ابتکارهایی که برای تغییر می‌زنید، مجبور نیستید از ابتدا شروع کنید. فقط کافی است چیزهایی را پیدا کنید که قبلاً جوابشان را پس داده‌اند، و با آن‌ها شروع کنید. خیلی روی نتیجه نهایی تمرکز نکنید، بلکه بیشتر روی رفتارهای اولیه تمرکز کنید. و روی همه آن‌ها هم نه، بلکه روی آن اندک رفتارهای حیاتی‌ای که باعث تفاوت زیادی می‌شوند. پیتر استیونس، یک مربی اسکرام، به من گفت هر وقت می‌خواهد مربی‌گری یک تیم جدید را شروع کند، از آن‌ها می‌پرسد: «بهترین پروژه شما کدام است؟» به گفته او، وقتی افراد بهترین تجربیاتشان را در یک فضای قصه‌گویی بیان می‌کنند، می‌توانند همان تجربیات مشترکی را که بقیه مردم آن‌ها را چابک یا ناب می‌نامند در آن تشخیص دهند. ولی او می‌گوید: «من به تیم‌ها نمی‌گویم که از اسکرام یا کانبان استفاده کنند. اگر پروژه فعلی‌شان بهترین پروژه‌شان نیست، من از آن‌ها می‌پرسم چطور می‌توان این پروژه را تبدیل به بهترین کرد؟ این روش به آن‌ها کمک می‌کند که تجربیات و روش‌هایی را که دوباره کشف کرده بودند به کار بگیرند.»

درخواست شما برای تغییر ممکن است برای خیلی از افراد سخت باشد. با الزام آن‌ها به قدم گذاشتن در قلمرویی ناشناخته آن را سخت‌تر نکنید. به آن‌ها نشان دهید که جایی که می‌خواهید بروید جای قشنگی است، و سپس روی اولین قدم‌ها توافق کنید.

انجام

اگر ندانید که از افراد می‌خواهید چه کاری انجام دهند، نمی‌توانید تغییری

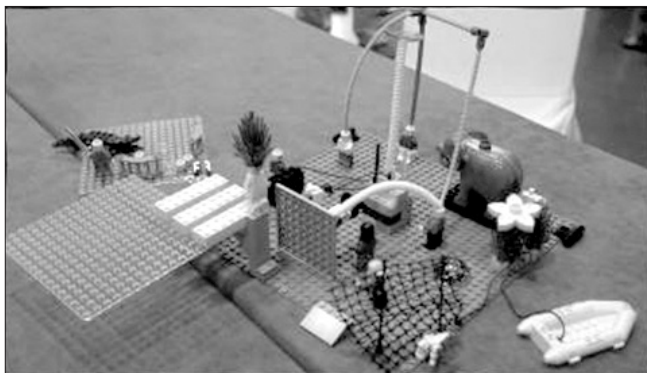
16- Lean

17- ALE network vision, <http://www.youtube.com/watch?v=Zg2PMv8IFUA>

13- Iterative

14- Metricate

15- Six Sigma, http://en.wikipedia.org/wiki/Six_Sigma



شکل ۴: چشم انداز چابک ناب اروپا^۱

1- Picture by Olaf Lewitz, Agile42, used with permission

چطور بازخورد بگیرید؟

به عنوان یک تغییر دهنده شما چیزهای زیادی را خواهید آزمود. شما باید بسنجید کدامها جواب می دهند و کدامها نه. به زبان دیگر، شما نیاز به بازخورد دارید. ولی چه کسی می تواند به شکل مثبتی نقد کند و به شما بازخوردی بدهد که واقعا به دردتان بخورد؟ شما باید افرادی را بیابید که نظراتشان با ارزش است. از آنها بخواهید ایده هایتان را سبک سنگین کنند و به حرف هایشان گوش کنید. تجربه هایتان را با کسانی انجام دهید که دوست دارند نتایج ایده های شما را بچشند و خوشحال می شوند که یافته هایشان را گزارش کنند، ترجیحا بدون این که آنها را توی صورتتان بکوبند.

وقتی من دوره آموزشی مدیریت ۳۰ را شروع کردم، چندین بار آن را با شنوندگان متفاوتی امتحان کردم. بار اول یک کارگاه پنج ساعته در گردهمایی اسکرام در آمستردام بود. بار دوم یک دوره رایگان یک روزه با حضور ۱۵ نفر بود. بار سوم یک دوره ۲ روزه کامل با حضور ۳۰ نفر از کارمندان یک مشتری بود. و هنوز هم پس از دهها دوره، من همچنان تجربه کردن و سنجش را تمام نکرده ام. من معمولا از شرکت کنندگان دعوت می کنم که قبل از ناهار یادداشتی روی یک در بگذارند (شکل ۶). چنین بازخورد کیفی به من کمک می کند که دوره را بهبود دهم و در جهت درست باقی بمانم.

چطور نتایج را اندازه بگیرید؟

یادداشت ها عالی هستند، ولی بازخورد کیفی اغلب کافی نیست. اگر نتایج کمی را اندازه نگیرید، ممکن است نفهمید آیا ابتکارات شما در مسیر درست پیش می روند یا نه. چطور باید این کار را انجام داد؟ چطور می توانید مفهومی انتزاعی مثل «چابک بودن» یا «سازمان یادگیرنده بودن» را اندازه بگیرید؟ خوشبختانه من پاسخ این پرسش ها را ندارم، چون پاسخ خیلی بستگی به تغییر مورد نظر شما دارد و چیزی است که شما باید خودتان پیدا کنید. همزمان شما باید مواظب باشید که معیارهای اندازه گیری تان منجر به رفتارهای ناکارآمد در سیستم نشود. به نظر دشوار می رسد؟ بله، چون واقعا هم دشوار است!

فرودگاه شیپول (در آمستردام) تعداد مسافرانی را اندازه می گیرد که در

ایجاد کنید. چطور می خواهید «رفتار خوب» را تعریف کنید؟ آن اندک حرکت های کلیدی که از افراد انتظار دارید کدامها هستند؟

قدم های کلیدی کدامها هستند؟

برای واداشتن افراد به حرکت، با قدم های کوچک شروع کنید. یکباره تصویر بزرگ را رویشان نیندازید، بلکه برایشان اهداف کوتاه مدت روشن و قابل دستیابی تعیین کنید. بله، یک چشم انداز بزرگ لازم است. ولی قدم های کلیدی برای حرکت در آن مسیر باید آن قدر ساده باشد که همه آنها را درک کنند.

در سال ۲۰۰۱، در پیست اسکی اسنوبورد در یوتا، ۱۷ نفر دور هم جمع شدند تا درباره روش های بهتر تولید نرم افزار صحبت کنند. آنها به چشم اندازی رسیدند که بزرگ و قانع کننده بود (شکل ۵). و آنها با گذاشتن نام «چابک» بر آن، متمایزش کردند و آن را تبدیل به یک بیانیه نمودند. ولی هریک از آنها جداگانه زمان زیادی را صرف یاد دادن تمرین هایی ساده و مشخص به افراد کردند، مثل «داستان ها را روی کاغذهای چسب دار بنویسید» و «کارهایتان را روی یک تخته سفید نمایش دهید» و «هر هفته از مشتری تان بازخورد بگیرید».

آغاز کنندگان موفق تغییر اغلب یک چشم انداز بزرگ را با تمرین هایی ساده و البته قاطع و مشخص همراه می کنند.

کی و از کجا شروع کنید؟

زمان و مکان برای آغاز هر تغییری بسیار مهم هستند. شما باید به دقت زمان درست و مکان درست برای شروع را پیدا کنید. گاهی وقت ها این یعنی باید برای شرایط مناسب تری صبر کنید. گاهی وقت ها هم به این معنی است که نباید بیشتر صبر کنید، و باید همین حالا انجامش دهید! مثلا انگولک کردن درست یک سیستم وقتی که در حال استراحت است و حساسیتش بالاست می تواند اثر شگفت آوری داشته باشد، ولی همان کار وقتی سیستم در حال کار است ممکن است هیچ اثری نداشته باشد.

دو سال تمام، من ایده ای درباره یک همکاری بین-اروپایی در زمینه های چابک و ناب را داشتم. ولی فقط بعد از انتشار کتابم بود که فهمیدم توجه مردم به من جلب شده و با افرادی در سراسر اروپا ارتباط پیدا کرده ام، و ناگهان دریافتم که زمان مناسب همان موقع است. باید فکر کردن درباره اش را متوقف می کردم و انجامش می دادم!

انجام دادن مدیریت تغییر شامل یافتن قدم های کلیدی که افراد باید بردارند، و انتخاب زمان و مکان درست برای شروع.

بررسی

سومین بخش چرخه PDCA درباره بازخورد است. وقتی به سیستم با ایده های جدید تلنگری می زنید، باید بررسی کنید و بفهمید که سیستم چطور پاسخ می دهد.



شکل ۶: بازخورد در طول یک دوره آموزشی

واکنش نشان دهم. من می‌توانستم به بازخوردها پس از نهار پاسخ دهم، و فقط چند ساعت بعد بازخوردهای جدیدی درباره تغییرات دریافت کنم. ولی شاید مهمترین جنبه بازخورد سریع، پتانسیل آن برای ایجاد یک چرخه بازخورد تقویتی است. دوره‌های کوتاه بازخورد به شما امکان می‌دهد از پیروزی‌های کوچک به‌عنوان گواهی بر با ارزش بودن تلاشی که برای تغییر صرف کرده اید استفاده کنید. وقتی توانستید بسنجید که ایده شما نتایج مفیدی دارد، می‌توانید به تدریج سرعت تغییر را بیشتر کنید و آن را در سازمان منتشر کنید. پیروزی‌های سریع می‌توانند یک چرخه بازخورد تقویتی ایجاد کنند که به شما کمک می‌کند دستاوردها را مستحکم کنید و تغییر مثبت تری انجام دهید.

من بحثی را از چند سال پیش به خاطر دارم درباره این که کدام نیازمندی‌ها را در یک پروژه انتخاب کنیم. برخی واقعا بزرگ و مهم بودند و برخی کوچک و کم اهمیت. باید کدام‌ها را انتخاب می‌کردیم؟ برای این که اولویت‌ها را بفهمیم، دو راه را کشیدیم که با هم ادغام شده و به یک راه منتهی می‌شدند. در یک مسیر اتوبوس‌ها حرکت می‌کنند (نیازمندی‌های بزرگ) و در مسیر دیگر دوچرخه‌ها (نیازمندی‌های کوچک). در نگاه اول ما فکر کردیم باید اول به اتوبوس‌ها اجازه عبور بدهیم. در نگاه دوم، معتقد شدیم در برخی موارد بهتر است اجازه دهیم چند دوچرخه عبور کنند. از نظر من این مانند انتخاب بین کایزن (تغییر مداوم در قدم‌های کوچک) و کایاکو (تغییر اساسی در قدم‌های بزرگ) است. خوب است نوعی حالت شهود برای هدایت انتخاب‌هایتان داشته باشید.

اندیشمند پیچیدگی باشید

یک سیستم اجتماعی، پیچیده و تطبیق پذیر است. به‌عنوان یک تغییر دهنده، باید به تلنگر زدن به آن با ایده‌های جدید و بررسی پاسخ و تغییر ادامه دهید. شما مجبورید به این تغییرات طوری پاسخ دهید انگار در حال رقصیدن با سیستم هستید و با دقت آن را در مسیر درست هدایت کنید. اگر شما برای ایجاد تغییر جدی هستید، باید پرسش‌های مهمی از خودتان بپرسید، مثل:

هدف من چیست؟

بیانیه توسعه جابک نرم افزار

ما در تلاش برای کشف راه های بهتر برای توسعه نرم افزار هستیم. با انجام آن، و با کمک به بقیه برای انجامش، با این کار ما به ارزش های زیر رسیده ایم:

افراد و تعاملات بهتر است از فرآیندها و ابزارها

نرم افزار کارکننده بهتر است از مستندات جامع

مشارکت مشتری در انجام کار بهتر است از قرارداد کاری

پاسخگویی به تغییرات بهتر است از پیروی از یک برنامه

با وجود این که موارد سمت چپ نیز ارزشمند هستند ولی ما برای موارد سمت راست ارزش بیشتری قائل هستیم

Kent Beck	James Grenning	Robert C. Martin
Mike Beedle	Jim Highsmith	Steve Mellor
Arie van Bennekum	Andrew Hunt	Ken Schwaber
Alistair Cockburn	Ron Jeffries	Jeff Sutherland
Ward Cunningham	Jon Kern	Dave Thomas
Martin Fowler	Brian Marick	

©2001, the above authors

This declaration may be freely copied in any form, but only in its entirety through this notice.

شکل ۵: بیانیه توسعه جابک نرم افزار

هر پرواز از اجناس معاف از مالیات خرید می‌کنند و این که مسافران از کجا آمده‌اند و به کجا می‌روند. من تعداد شرکت کنندگانی را که مایلند ارزیابی شان را به من بدهند اندازه می‌گیرم و این که کدام قسمت‌های دوره را بیشتر دوست داشته اند و کل دوره را چطور ارزیابی می‌کنند. یک تغییر دهنده، چیزهایی را اندازه می‌گیرد تا بتواند همبستگی و ارتباط نتایج و ایده‌ها را بیابد. ما چه کاری کردیم که باعث بروز یک تغییر خاص شد؟ آیا نتیجه خوب بود؟ اگر نه، آیا ما می‌توانیم روش دیگری را بیازماییم؟ در کتاب استارت‌آپ ناب، اریک ریس این کار را «یادگیری معتبر»^{۱۸} می‌نامد [Ries 2011]. این یعنی اندازه گرفتن این که کدام یک از دخالت‌ها واقعا اثر درستی روی افراد داشته (و کدام‌ها نه).

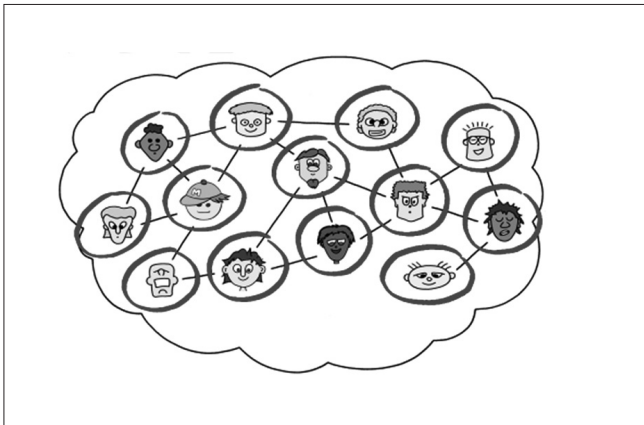
کنش

دریافت بازخورد در دوره‌های کوتاه بهتر از دوره‌های بلند است. دانستن این که چیزی جواب می‌دهد یا نه، جلوی صرف هزینه زیاد در مدت طولانی روی یک روش بد را می‌گیرد.

چطور به نتایج سرعت بدهید؟

بازخورد سریع ریسک خطا را کاهش داده، فرصت بیشتری برای تجربه در اختیارتان قرار می‌دهد. و وقتی هنوز مسائل در ذهن افراد تازه هستند، آن‌ها می‌توانند جزییات بیشتری را در بازخوردهایشان به شما منعکس کنند.

یکی از دلایل این که من از شرکت کنندگان می‌خواهم که چندین بار در طول دوره نظراتشان را روی برگه‌های یادداشت بنویسند این است که بازخورد آن‌ها را پیش از آن که آن را فراموش کنند بگیرم. اوایل من این نظرات را تا وقتی که به خانه برگردم نمی‌خواندم. (اگر آن‌ها کارهای من را دوست نداشتند چه؟ ترسناک است!) ولی به زودی شروع کردم به خواندن آن‌ها در زمان‌های استراحت، طوری که بتوانم در صورت امکان به سرعت



شکل ۷: افراد در یک شبکه‌ی اجتماعی

۷). و از آنجا که آدم‌ها با هم فرق دارند، نمی‌شود برای تغییر اجتماعی یک راه حل ارائه کرد که به درد همه بخورد. اگر می‌خواهید یک سازمان بهبود یابد، باید به نیازهای تک افراد و موانع متفاوتی که آدم‌ها در ذهنشان دارند بپردازید.

طراحان در می‌یابند که در بیشتر سازمان‌ها می‌توانند در چارچوب همان محدوده طراحی شان تا حد قابل ملاحظه‌ای بهبود یابند. این به آن خاطر است که موانع تغییر عموماً در ذهن خود تصمیم‌گیرها و در خود سازمان است.

اگر شما می‌خواهید سازمانی را در سطح افرادش تغییر دهید، می‌توانید یک جراح مغز و اعصاب استخدام کنید. ولی ارزان‌تر (و موثرتر) آن است که مدل ADAKR (که توسط جف هیات ارائه شده) را به کار بگیرید [Hiatt 2006].

شکل ۸ یک مدل مدیریت تغییرات هدف‌گراست که برای هدایت فعالیت‌ها در طی دوره تغییر ایجاد شده است. مدل به این درد می‌خورد که اطمینان حاصل کنید در ابتکاراتی که برای تغییر می‌زنید، جنبه انسانی موضوع را فراموش نمی‌کنید. مدل ADAKR این پنج بعد را دارد:

آگاهی از نیاز به تغییر

میل به مشارکت در تغییر و پشتیبانی از آن

دانش درباره‌ی چگونگی تغییر (و این که تغییر چگونه چیزی است)

توانایی پیاده‌سازی تغییر به شکل روز به روز

تقویت برای نگه داری تغییر

فایده‌ی مدل ADAKR در این است که به شما یادآوری می‌کند تغییر اشخاص در یک سازمان نیازمند توجه به حوزه‌های متعددی است. فقط آگاهی افراد از نیاز به تغییر کافی نیست. آن‌ها باید میل به تغییر هم داشته باشند.

میل به تغییر هم کافی نیست، اگر افراد دانش چگونگی تغییر، یا توانایی آن را نداشته باشند.

و حتی اگر آن‌ها بدانند چگونه، و توانایی تغییر را هم داشته باشند، بدون تقویت رفتارهای خوب، افراد به عادت‌های قدیمشان باز می‌گردند.

کجای کار خوب پیش می‌رود؟
قدم‌های ابتدایی کدام‌ها هستند؟
کی و کجا شروع کنیم؟
چطور باز خورد بگیریم؟
چطور نتایج را اندازه بگیریم؟
چطور به نتایج سرعت بدهیم؟

به آدم‌ها اهمیت بده

بیشتر آدم‌ها، آدم‌های دیگرند.
افکار آن‌ها عقاید یک نفر دیگر است،
زندگی‌شان یک تقلید،
احساسات‌شان نقل قول.

اسکار وایلد، نویسنده ایرلندی (۱۸۵۴-۱۹۰۰)

هدف ما در گردهمایی استوس این بود که به درکی از مشکل موجود در سازمان‌های مدیریت شده در سراسر دنیا برسیم، با توافقی درباره‌ی آینده مطلوب، و حداقل پیشنهادهایی برای سریع‌تر رسیدن با آن نقطه.

شرکت کنندگان مشکل را به شکل یک «نقشه ذهن پنجاه بعدی»^{۱۹} (نامی که استیو دنینگ برایش گذاشت) تعریف کردند. این نقشه از نظر ریشه‌های مسائل و عواقب آن‌ها نقشه فراگیری بود. ولی عمداً آن را «کار ناتمام» نامیده بودیم. چون همان‌طور که در فصل پیش دیدیم، مشکلات هم همیشه در حال تکامل هستند.

گروه، آینده ایده آل را این‌طور تصویر کرد:

سازمان‌ها می‌توانند تبدیل شوند به شبکه‌هایی یادگیرنده از افرادی که ارزش ایجاد می‌کنند، و نقش رهبران باید پاسداشت^{۲۰} زندگی باشد، نه مدیریت ماشین.

ممکن است بعضی‌ها بگویند این هدف بزرگ قابل اندازه‌گیری نیست، و درست هم می‌گویند. ولی به نظر^{۲۱} شرکت‌کننده در این گردهمایی، اعلامیه نهایی به خوبی امیدها و آرزوهای آن‌ها را منعکس می‌کند و آن‌ها امیدوارند بقیه هم آن را الهام بخش بیابند.

صدالبته به‌عنوان عوامل تغییر، ما می‌دانیم که پیشرفت واقعی با قدم‌هایی کوچک و قابل اندازه‌گیری انجام می‌شود. و برای انجام آن، گروه شرکت‌کننده در استوس درباره‌ی افراد سودبر (دینفع) و اثرگذار و راهبردها هم بحث کردند. و ما بر نیاز به پاسخ به این سوال تاکید کردیم که «برای من چه دارد؟» و «چطور می‌توانم انجامش دهم؟» تغییر رخ نمی‌دهد، وقتی مردم آن را به‌عنوان چیزی می‌بینند که برای خودشان فایده‌ای ندارد، یا چیزی که آن‌ها قادر به اجرایش نیستند. به همین دلیل است که ما در این فصل درباره‌ی چگونگی انجام تغییر در سطح تک تک افراد صحبت می‌کنیم.

موانع را بردارید

مهم‌ترین اجزای یک سیستم اجتماعی افراد حاضر در آن هستند (شکل

19-Stoos Network problem definition, <http://www.stoosnetwork.org/what-is-the-problem/>
20- Stewardship



شکل ۹: آیا این پیام شما را قانع می‌کند که سیگار نکشید؟

چطور یک مثال فراهم می‌کنید؟

آدم با دیدن باور می‌کند. هیچ نیروی انگیزشی بهتر از داشتن رهبری نیست که با نمونه و مثال رهبری می‌کند. برای بیشتر افراد، آن چه شما انجام می‌دهید بسیار مهم‌تر است از آنچه شما می‌گویید. بنابراین، شما نباید افراد را تنها با گفتن آن چه می‌خواهید مطلع کنید. بلکه همچنین باید با عمل کردن به موعظه‌هایتان، به آن‌ها نشان بدهید چشم انداز شما چطور اجرا می‌شود.

من توانستم با به کار گرفتن شیوه‌های چابک در کارهای روزانه خودم، به سازمانم کمک کنم که چابک‌تر باشد. بقیه افراد وقتی آن چه را که من انجام می‌دهم می‌بینند، آسان‌تر نیاز به تغییر را می‌پذیرند.

آگاه کردن افراد از نیاز به تغییر نیازمند ارتباطات خوب است که چیزی آموختنی است و نیز نیازمند مثال‌هایی مناسب که بقیه بتوانند از آن‌ها پیروی کنند.

میل

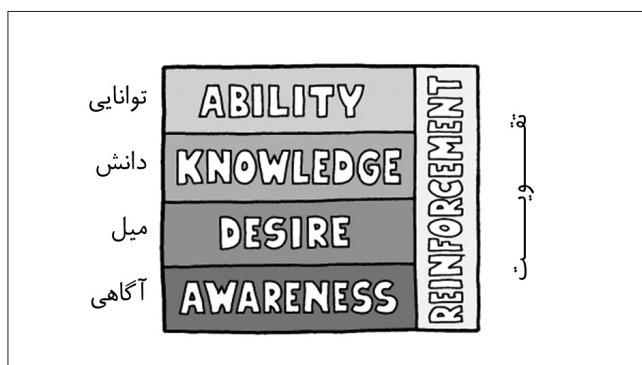
ایجاد آگاهی اولین قدم به طرف تغییر است. ولی شما به عنوان یک تغییر دهنده باید بدانید که هیچ گاه فقط آگاه کردن مردم از ایده تان کافی نیست. پیام منطقی شما باید با محرک‌های احساسی برای تغییر همراه شود.

چطور آن را ضروری می‌کنید؟

خیلی وقت‌ها افراد چیزهای مهم را نادیده می‌گیرند و فقط روی چیزهای ضروری تمرکز می‌کنند [Covey 2004]. متأسفانه خیلی از مسائل خودشان را به شکل‌های ضروری نشان نمی‌دهند. و بنابراین، اتفاقی نمی‌افتد. مردم ممکن است اهمیت ایده شما را درک کنند، ولی ضرورت آن را احساس نکنند. بنابراین برای واداشتن مردم به تغییر، شما باید تلاش کنید تا آن‌ها بحران را حس کنند، و ایده شما را به عنوان چیزی ضروری (و نه تنها مهم) تجربه کنند.

پدر من می‌دانست که خوردن غذاهای سالم مهم است، ولی این مسئله هیچ وقت برایش ضروری و حیاتی نبود. تا وقتی که به دلیل احساس ناخوشی نزد دکتر رفت. دکتر نارسایی شدید قلبی او را تشخیص داد و به او گفت که تنها چند ماه دیگر زنده خواهد ماند، مگر این که بلافاصله و به شدت رژیم غذایی اش را تغییر دهد. حالا ضرورتش حس می‌شد! و بنابراین او تغییر کرد.

ادامه دارد



شکل ۸: مدل - آگاهی، میل، دانش، توانایی، تقویت

وقتی یک حرکت برای تغییر را شروع می‌کنید، ممکن است بخواهید از مدل ADAKR برای در نظر گرفتن تغییر در سطح افراد در هر پنج بعد استفاده کنید و موانع ذهنی افراد را بردارید. در ادامه این فصل این مدل را توضیح می‌دهم.

آگاهی

برای حرکت دادن آدم‌ها به جلو در برنامه تغییرتان، معمولاً یک یادآوری ساده که به آن‌ها بگوید «از اول هفته آینده، کارها را به شکل متفاوتی انجام دهید» کافی نیست، مستقل از این که از نظر شما تغییر چقدر مهم است.

چگونه ارتباط برقرار خواهید کرد؟

بیشتر مردم در انتقال ایده‌های شان قوی نیستند. باراک اوباما رییس جمهور آمریکا نمی‌شد اگر من می‌کرد، کلمات دور از ذهن به کار می‌برد، به زمین نگاه می‌کرد، و حاضرین را با ده‌ها نکته شماره گذاری شده می‌کشت. طرفدارانش هم از پوسترهایی با حروف ریز، با دستخط بد، با جوهر کمرنگ، روی کیف‌های کاغذی قهوه‌ای استفاده نمی‌کردند. من با کمال تعجب دیده‌ام که عاملان تغییر تلاش می‌کنند پیام شان را درست به همین روش‌هایی که من توضیح داده‌ام برسانند.

شما باید به طور مداوم و به روش‌های متفاوت پیام تان را تکرار کنید. من آموخته‌ام که کمی سرمایه گذاری در مهارت‌های ارتباطی، بازده خوبی خواهد داشت. مثلاً کتاب ساخته شده برای چسبیدن^{۲۱} به شما روش‌هایی برای انتقال پیام با داستان‌هایی ساده، نامنتظر، مستحکم، و احساسی یاد می‌دهد [Heath 2007].

همه بسته‌های سیگار در کشور من جیغ می‌زنند که «سیگار کشیدن باعث مرگ می‌شود»، ولی این پیام روی بیشتر مردم اثری ندارد (شکل ۹). این بدان خاطر است که انتقال پیام اولین قدم به سوی تغییر است. کافی نیست، ولی نقطه آغاز است. شما باید بیاموزید که با دقت و توجه و تعمق به ذهن مردم رسوخ کنید و باید قصه‌ها و تصویرهای خوبی برای پشتیبانی از ایده تان داشته باشید.

وقتی به یک سیگاری گفته می‌شود که سیگار کشیدن را متوقف کند، نتیجه‌اش عموماً این است که یا ادامه می‌دهد و یا مقدار سیگارش را افزایش می‌دهد.

زنگ تفریح

۱۶ روز

چند تا مهندس در کافی شاپ دور هم نشسته بودند و با هم با صدای بلند می‌گفتند: «۱۶ روز» یکی از خدمتکاران کافی شاپ، ضمن آن که آن‌ها را دعوت به سکوت می‌کرد، پرسید جریان ۱۶ روز چیست؟ یکی از مهندسان گفت ما با هم توانستیم یک پازل را فقط در ۱۶ روز سر هم کنیم و برای همین جشن گرفته‌ایم.

خدمتکار گفت: حتماً پازل خیلی سختی بوده است؟ مهندس جواب داد: البته، البته، پشت جعبه‌اش نوشته بود «۵ تا ۷ سال»

اعداد اوّل

به یک ریاضیدان، یک فیزیکدان، یک مهندس و یک برنامه‌نویس کامپیوتر مسئله‌ای داده شد و از آن‌ها خواسته شد که پاسخ را بدهند.

مسئله این بود: آیا تمام اعداد فرد، اوّل هستند؟ همه آن‌ها رفتند تا جواب مسئله را پیدا کنند.

ریاضیدان اوّل از همه برگشت. او گفت ۳ اوّل است، ۵ اوّل است، ۷ اوّل است، ۹ اوّل نیست. پس پاسخ سؤال، خیر است. نفر دوم فیزیکدان بود. او گفت: ۳ اوّل است، ۵ اوّل است، ۷ اوّل است، ۹ اوّل نیست، ۱۱ اوّل است، ۱۳ اوّل است. پس ۹ خطای آزمایش بوده و پاسخ سوال مثبت است.

پس از مدتی مهندس برگشت. او گفت: ۳ اوّل است، ۵ اوّل است، ۷ اوّل است، ۹ اوّل است. پس پاسخ سؤال مثبت است.

برنامه‌نویس کامپیوتر وقتی جواب‌های آن سه نفر را شنید، گفت: حالا که شما نتوانستید بر سر جواب توافق کنید. من برنامه‌ای نوشته‌ام که جواب قطعی را خواهد داد. سپس برنامه‌اش را از طریق صفحه کلید وارد کامپیوتر کرد و دکمه «Enter» را زد.

برنامه بلافاصله اجرا شد و خروجی آن به ترتیب زیر روی صفحه نمایش ظاهر شد:

۳ اوّل است

۳ اوّل است

۳ اوّل است

ویژگی بعضی مدیران

بعضی مدیران مثل مرغ دریایی می‌مانند: ناگهان از آسمان سر می‌رسند، سروصدای زیادی تولید می‌کنند، به همه چیز گند می‌زنند و بعد آنجا را ترک می‌کنند.

توصیه‌های کاری

- هرگز در داخل برنامه‌هایتان جملات توضیحی (Comment) که به دیگران کمک کند تا از کد شما سر در بیاورند، قرار ندهید. اگر آن‌ها از برنامه شما سر در بیاورند دیگر به شما نیازی نخواهند داشت.
- هرگز پروژه‌ای را سر وقت تمام نکنید. اگر چنین فکر می‌کنند که کار ساده‌ای بوده که هر کس می‌توانسته انجامش دهد و دیگر به شما نیاز نخواهند داشت.
- هنگامی که کسی به دفتر شما آمد و سؤالی پرسید، پشت سر هم حرف بزنید و مجال صحبت کردن به او ندهید ولی جوابش را هم ندهید. اگر پاسخ سوال هایشان را بیابند دیگر به شما نیازی نخواهند داشت.
- هرگز برنامه‌هایی که نوشته‌اید را در دسترس دیگران نگذارید و آن‌ها را در کشوی میزتان قایم کنید. اگر آن‌ها به برنامه‌های شما دسترسی داشته باشند دیگر به شما نیازی نخواهند داشت.
- هرگز در مورد هیچ چیز به طور روشن و واضح صحبت نکنید. همیشه دو پهلو و انتزاعی صحبت کنید. اگر آن‌ها بفهمند که شما چه می‌گوئید دیگر به شما نیازی نخواهند داشت.

لیست اعضای حقوقی

انجمن انفورماتیک ایران

اعضای حقوقی فعال در حوزه راه حل های برنامه ریزی منابع سازمان و نرم افزارهای پیشرفته سازمانی

آریا همراه سامانه



آفرینش رایانه کیهان



ایریسا (بین المللی مهندسی سیستم ها و اتوماسیون)



بهکو (بهینه کوشان سپهر)



برگ سیستم پویا



پارس رویال نفیس



پارس اوک کیش



پارس تصمیم



پردازش موازی سامان



پردازش اطلاعات ریسمان



پویندگان تجارت دهکده جهانی



توسعه یکپارچه ایلیا



تدوین فرآیند



تحقیق و توسعه ارتباط



چارگون



خدمات مهندسی عصر اندیشه



درگاه ارتباطات جدید



دی رایانه نوین



دانش افزار نارون شریف



راهکارهای برتر یکپارچه ساپارد اروین



رایاوران توسعه



مهندسی رای دانا آفرین



سامانه پرداز اریس



پایه ریزان راه کارهای فراگیر



سامه آرا پردازشگر



سبز داده افزار



سیستم های اطلاعات مدیریت شرق رایا



سند پرداز



شماران سیستم



گروه نرم افزاری پیوست



گلرنگ سیستم



فناوری نوین رایا شریف



فرهیختگان فناوری اطلاعات و ارتباطات
فرافن



فناوری اطلاعات و ارتباطات پاسارگاد
آریان (فناپ)



فاوا نفت صبا کیش



همکاران سیستم مدیریت طرح های
عمومی



نوید ایرانیان
(گسترش فضای مجازی)



مانیر (مشاورین انفورماتیک نیرو)



مجمع داده ها و سیستم ها (MDS)



مهندسی بهینه ایران



ماموت مدیریت سیستم ها



مهندسی نرم افزاری رایورز



مهندسی نرم افزار فرارای (سهامی خاص)



نماد ایران



نسیم ارتباط دماوند

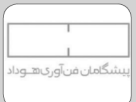
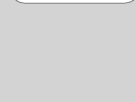


داده پردازی نیلرام مانا



لیست اعضای حقوقی

انجمن انفورماتیک ایران

دی کاو ماندگار		پیک فرزانه پویان		گروه مشاورین و رانگر نوین	
راهبر نیروی خراسان (رانیر)		پاریزان صنعت دانش		گروه توسعه فناوری اطلاعات بصیر	
طرفه نگار		پیشگامان فن آوری هوداد		گام الکترونیک	
رایانه خدمات امید		توسعه ارتباطات رایانه‌ای آبانگان		مشاوران یام آذر	
سافت سیستم کیش		توسعه فناوری اطلاعات جهان افزارنوین		نوماتک	
سنجه حساب		تدبیرگران نوآوری رایسان		اعضای حقوقی فعال در حوزه نرم افزارهای کاربردی	
سبزافزار آریا		توسعه زیر ساخت ماهان تک			
سروش رایانه ایرانیان		تیم تعمیرات و نگهداری خطوط لوله خاورمیانه		اطلاع رسانی پیوند داده‌ها	
شایگان سیستم		تحلیلگران فن آوری پوشش نور		ایران رایانه	
صنایع فن آوری طراحان بهینه		حامیان ماندگار فرتاک روشن		ایده نوآوران سرا تدبیر	
مهندسی فرا طرح نوین مهرگان پیشتاز		حافظه سبز نوین پارس		امن پردازان سورنا	
مهندسی تکرر سیستم		خدمات مهندسی فن آوری های طیف		آریانا پردازان آینده (آریا)	
مدیران سیستم پاسارگاد		چشم انداز تجارت به دان		انتقال دانش صنعت انرژی برق	
مهندسی نرم افزار ایده گستر پوریا		داده پردازان احداث		پندار کوشک ایمن	
مهندسی نرم افزار فرا پیام		داده پردازان آبشار		پژند الکترونیک	
مهندسین مشاور نقش بوستان گستر		داده پردازی حرفه‌ای‌ها		پردازش اطلاعات و ارتباطات هاماوران آسیا	

لیست اعضای حقوقی

انجمن انفورماتیک ایران

توسعه خدمات الکترونیکی آدونیس



ثامن ارتباط عصر



شرکت کسب و کارهای نوپای خاورزمین



ماتریس تحلیلگران سیستم‌های پیچیده



خدمات انفورماتیک



توسعه سامانه‌های نرم‌افزاری نگین (توسن)



خدمات ماشین‌های اداری امیم رایانه



سامان امنیت پردازش



سفیر آبی آرام



شرکت توسعه فن‌افزار توسن



داده‌پردازی ایران



داده‌پردازان ماهان شبکه جنوب



داده‌ورزی فرادیس البرز



داتیس آرین قشم



شرکت شبکه الکترونیکی پرداخت کارت شاپرک



سامانه تبادل الکترونیک دفاتر پیشخوان دولت



اعضای حقوقی فعال در حوزه بانکی و بانکداری الکترونیکی

الماس هوشمند ایرانیان



آسمان صبح فردا



ارتباطات هدی ارقام



بهبازان ملت



بانک اقتصاد نوین



بانک آینده



به‌پرداخت ملت



پدیسار انفورماتیک



پردازشگران سامان



پرداخت الکترونیک سداد



پویا



بانک سرمایه



توسعه فن‌افزار توسن



توسعه فناوری رفاه پردیس



توسعه فناوری تجارت حکمت



فاوا هزاره کیش



فراکنش



فناوری ارتباط امن خاورمیانه



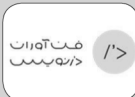
همراهان سیستم گوهر



همکاران سیستم



فن اوران درنویس



نرم‌افزاری امن پرداز



رهیاب ریان فردا



کهکشان دانا



مدار گسترش فناوری اطلاعات



موسسه فرهنگی دیجیتال تدارک نرم‌افزار پایتخت پارسیان



کارانس ایرانیان



گروه شرکت های مهندسی نرم افزار فراپیم



گستره چتر نیلی



حامی سیستم شریف



لیست اعضای حقوقی

انجمن انفورماتیک ایران

کیانا پارسیان کیش



کلیدگستر آینده (نت بینا)



گروه شرکت های فن آوا



مؤسسه گسترش اطلاعات و ارتباطات
و فرهنگی ندا رایانه



هاست ایران



نویان ابر آوران



اعضای حقوقی فعال در حوزه شبکه و سخت افزار

آوید رایانه هیراد



آداک فناوری مانیا



آلبالو رایانه سخت افزار



آرین وب ایرانیان



ایده نگر نوین اطلس



پارس فایبرنت



پیشتاژان اندیشه پویا



اروند رایان گستر



برد پرداز رایانه



ارتباطات مبین نت



افرانت



پارس لین



پیشگامان دامنه فناوری



توسعه آرمان ارتباطات تاریانا



تندیس تلاش و تفکر



توسعه فناوری ریز فن پردازش پرشین



تبادل الکترونیک فردا



تدبیر (ایران سون استار)



فناوران اهل قلم نوین



داده پردازی پویای شریف



دانشگاه آزاد اسلامی واحد سوسنگرد



رایان هوشمند نویان



فناوری اطلاعات و ارتباطات پارس



پردازش سدید



فن آوری اطلاعات پارسیان میزبان



فناوری رادین پاسارگاد

فناوران تجارت الکترونیک
الماس



مشاورین بهبود روش ها و
سامانه های مبنا



فناوری اطلاعات ناواکو



فرانگر صنعت کارت اریا



فناوری اطلاعات ارتباطات پارسه بین
الملل اتیه ونداد



مدیریت و مهندسی کرون راد



گرایش تازه کیش



گسترش فناوری های نوین کشاورز



مهندسی آتی ساز اب ایرانیان



توسعه فناوری اطلاعات خوارزمی



بانک سینا



هوشمند الکترونیک
الماس شرق کیش



اعضای حقوقی فعال در حوزه اینترنت و پورتال ها

آریا رسانه نوین تکنابن



آوای هوشمند زورق آرامش



ارتباطات فن آوا



لیست اعضای حقوقی

انجمن انفورماتیک ایران

شایان توسعه البرز



شیوه نوین ارتباط پایدار



شبکه پایدار فناوری اطلاعات



شبکه هوشمند پدیده آپادانا



صبا کاربردلتا



شرکت مهندسی گسترش ارتباطات نو خاورمیانه



شبکه گستر نسل جدید



فن آوری اطلاعات و ارتباطات
آرمان تندیس ایرانیان (فن آرا)



فناوران آدرین داده پرداز



کارنما رایانه



کیسان صنعت ایرانیان



کیسون



گروه مهندسیین فرایند



گروه بازرگانی آریانا پارس راژمان



گروه داده ورز جویا



لایف سرویس پارسه



خدمات رایانه ای بهینه پرداز پویا



خدمات آواژنگ



چرخه ارتباط سبز



سرو رایانه



سامانه هوشمند سورنا



سازگار الماس شرق



سیمرغ سامانه تهران



شکوه تجارت صدف



فاوا موج



فناوری اطلاعات سهلان



فرا اتصال رایان عصر



فرصت کیش



فدک رایان



مهندسی پیمان عمران نیرو



مهندسی رادین راهبرد رایانه



موسسه آموزشی نکوشی



پارسیان فیبر ارتباط



پرتو بیتا جاوید



پیشرو توسعه نوآوران آرتا



توانش



ترویج صنعت سومی پارسیان



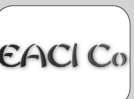
تحلیگران شبکه گستر پارسه



تحلیگران اطلاعات نگاره



تحلیگران ارتباط ایرانیان



توسعه سرمایه گذاری گروه آروند



توسعه داده پردازی بنیس



دانشگاه آزاد علوم تحقیقات
خوزستان



دانش رایان ارتباطات روز



رایان مهر الکتریک



رهنمون فناوری اطلاعات



رهروان طب ثمین



خدمات کامپیوتری خانه سیستم



لیست اعضای حقوقی

انجمن انفورماتیک ایران

خدمات مشاور خرد پیروز



زورق آرامش



مهندسی و ساخت بویلر مپنا



گروه کارخانجات چینی مقصود



گسترش فناوری اطلاعات



ستاره ویدا

مهندسی آریا تدبیر ایلیا



طبیعت زنده

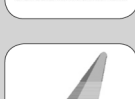


اعضای حقوقی فعال در حوزه آموزش و پژوهش

آماج فناوری اطلاعات



وارتباطات ایرانیان



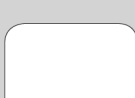
آزمایشگاه آزمون و تایید نرم افزار



آموزشگاه دخترانه سما- واحد کرج



آموزشگاه فنی و حرفه ای سما



الماس رایان ایرانیان



سماتک



فناوری اطلاعات بین الملل



مؤسسه عالی مدیریت دانش



اندیشه وران



سامان اندیشه ستیا



نفت ایرانول



اعضای حقوقی مشتریان بهره‌بردار از راه‌حل‌های نرم‌افزارهای پیشرفته سازمانی و بهره‌بردار از فناوری اطلاعات

آوای فناوری اطلاعات سلامت



ارتباطات نت میهن



اتصال صنعت میانه



تعاونی آموزشی خدماتی رایانه ای پیام دانش شهرکرد



بهسازان صنعت و اطلاعات



پارسین تجارت پایا کیش



پتروسان کالا نماد



توزیع برق آذربایجان



توسعه راهکارهای نور مبین (ترنم)



توسعه زیر ساخت سفر



ترابری بین المللی پرس



پردازش هوشمند البرز



مهندسی رایان توان افزار



مهندسی شبکه گستر



مهندسی پارتیان ابتکار پایدار



مهندسی افق داده ایرانیان



نوین ارتباط نوآوا



شرکت توانش



اندیشه پرداز دوران



گروه تجارت اوزن کیش



انتقال داده پرشین



ویژن پلاس اروپا



ویرا صنعت پارسه



نواوران اندیش رایانه غرب



همجو رایانه تهران



اعضای حقوقی فعال در حوزه مدیریت پروژه

آتی سپهر توسعه آریاناوآر



فناوران اطلاعات بهاران



لیست اعضای حقوقی

انجمن انفورماتیک ایران



سبزه داده افزار

نام مدیرعامل: سید وحید صدرالدینی
خدمات: ارائه دهنده راه حل های یکپارچه و تولید کننده ERP
برای سازمان های پروژه ای، تولیدی، بازرگانی و فروشگاهی
تلفن: ۲۲۲۸۹۲۵
نمابر: ۲۲۲۵۲۳۳۷
ایمیل: info@greendataware.com
آدرس وب: www.GreenDataware.com
آدرس: میرداماد، میدان مادر، خیابان شاه نظری، پلاک ۲۸، برج ناهید، طبقه ۱۲



شرکت نت بینا

- عضو انجمن انفورماتیک ایران
- عضو سازمان نظام صنفی رایانه ای کشور
- عضو سازمان جهانی IAA - سازمان iab

مدیرعامل: سرکار خانم طاهره خانی کلوری
خدمات: تبلیغات اینترنتی
توضیحات: نت بینا تنها نماینده رسمی شرکت پرسیکام انگلیس در ایران از سال ۱۹۹۵ در زمینه فعالیت های تحت وب مشغول می باشد این شرکت با به کارگیری تیم متخصص و حرفه ای در زمینه تبلیغات اینترنتی و راه حل های بازاریابی اینترنتی به شرکت های بزرگ ایرانی و خارجی در حال خدمت دهی است.
تیم حرفه ای نت بینا تاکنون به شرکت های بزرگ مانند: سامسونگ، ال جی، کانن، اچ تی سی، نسله، هواپیمایی امارات، سونی اریکسون، کاله، مبین نت، ماهان، نت برگ، مانگو، شیپور، سرویس های ویژه و اختصاصی جهت پیشبرد هدف های تبلیغاتی خود داده است.
تلفن: ۸۸۰۴۵۰۰۲
نمابر: ۸۸۶۰۰۵۴۰
ایمیل: sales@netbina.com
وبسایت: www.netbina.com
آدرس: شیخ بهایی شمالی، میدان پیروزان، خیابان پیروزان، نبش کوچه بیست و یکم، پلاک ۱۸، ساختمان نت بینا، واحد ۶



• عضو انجمن انفورماتیک ایران

مدیرعامل: یحیی پالیزدار
رئیس هیات مدیره: محمد حسین معینی
مدیر فنی: حسن نیلفروشان
خدمات: ارائه دهنده راه حل های یکپارچه انفورماتیکی و تولید کننده ERP
تلفن: ۸۸۵۴۸۸۰۲
نمابر: ۸۸۵۴۸۶۷۶
ایمیل: info@raydana.com
وبسایت: www.raydana.com
آدرس: شهرودی شمالی، خیابان قندی (پالیزی)، ابتدای خیابان سبزه دهم، شماره ۱۱



شرکت آریا همراه سامانه

- عضو نظام صنفی رایانه ای کشور
- عضو سندیکای صنعت برق
- عضو انجمن انفورماتیک ایران

نام مدیران: جناب آقای دکتر ثنائی (مدیرعامل و عضو هیات مدیره)
جناب آقای مهندس معدنی (مدیر راهکارهای یکپارچه نرم افزاری ERP)

خدمات: مشاوره و پیاده سازی نرم افزارهای جامع سازمانی (ERP)، ارائه خدمات راهبری و پشتیبانی مراکز داده، پروژه های شبکه و انتقال داده، تولید راهکارهای پیشرفته بومی امنیت شبکه و داده، پروژه های زیرساخت پیشرفته اندازه گیری انرژی (AMI) در شبکه های هوشمند، پروژه های جریان ضعیف در مترو و خطوط ریلی، توسعه مراکز داده (Data Center)، ارائه خدمات محتوا و سرویس های ارزش افزوده (VAS)، زیرساخت های تجارت الکترونیک و ارائه سرویس
تلفن: ۲۲۶۳۰۶۷۰-۸۰
نمابر: ۲۲۶۳۰۶۸۴
ایمیل و سایت: www.aryahamrah.com
info@aryahamrah.com /
آدرس: تهران، بزرگراه شهید مدرس، خیابان ظفر شرقی، خیابان فرید افشار، بلوار آرش غربی، پلاک ۹



شرکت ارتباطات مبین نت

- عضویت در انجمن انفورماتیک ایران

□ برترین اپراتور وایمکس در ایران به انتخاب وزارت ارتباطات و فناوری اطلاعات
مدیرعامل: حسین ریاضی
زمینه فعالیت: ارائه دهنده خدمات ارتباطی باندپهن بی سیم ثابت با فناوری های WiMAX, LTE, Wi-Fi در بخش های خانگی، اداری و سازمانی در سراسر ایران
تلفن: ۰۲۱ - ۸۳۸۶۹۰۰۰
فاکس: ۰۲۱ - ۸۸۷۰۰۳۹۶
تلفن مرکز خدمات: ۱۵۷۵
وبسایت: www.mobinnet.ir
پست الکترونیک: info@mobinnet.ir
نشانی دفتر مرکزی: تهران - ابتدای خیابان استاد مطهری - پلاک ۳۷۷

جای تبلیغ شما خالی است
۸۸۳۲۸۴۱۷-۲۱

کهکشان نور



شرکت علم یاران



پرداز گستر تدبیر



طلیعه کاوش



گروه فنی مهندسی ایده سازان



مبتکر قرن



موسسه پژوهشی پندار همایش



اعضای حقوقی فعال در حوزه مشاوره

آرین



الهه کوچک نرم افزار



تیوا سیستم



مشاوره مدیریت رایزن سامانه گستر



موسسه حقوقی و اقتصادی آرمان ایرانیان



طرح و توسعه الکترونیک افق



سرآمد فناوری اطلاعات



مشاوره مدیریت پیشداد سرویس



مهندسی کاربرد سیستم سدید

