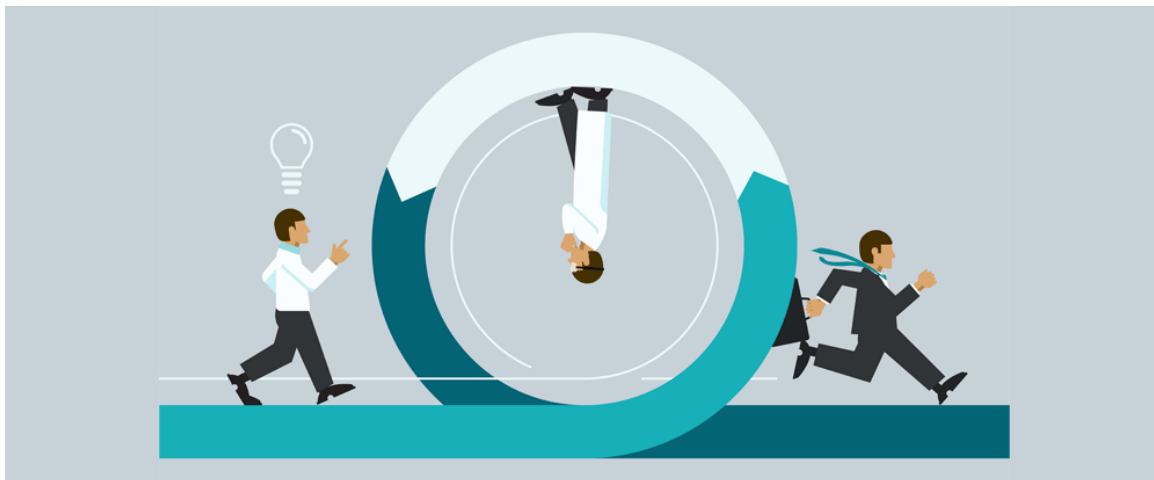


متدولوژی چابک در خدمت پروژه‌های تحول دیجیتال ۴۲



نویسنده: هنی پورتمن (Henny Portman)

ترجمه و ویرایش: www.Mostafa-Amini.com

چند سال پیش شما می‌توانستید بگویید که «اسکرام، چابک است» و بپرسید که «آیا چابک اسکرام است؟» هم اکنون می‌دانیم که بحث بسیار فراتر از اینهاست.

در حال حاضر بیش از ۵۰ رویکرد، چارچوب و روش چابک شناخته شده و ناشناخته وجود دارند. چابک بسیار فراتر از یک متدولوژی واحد است. در واقع می‌توان بحث کرد «متدولوژی چابک» به خودی خود اصلاً وجود ندارد. اغلب متخصصان پروژه می‌گویند که سازمان آنها متدولوژی‌های مدیریت پروژه را بصورت ترکیبی پیاده‌سازی کرده است که بسیاری از آنها اصول چابک را با روش‌های دیگری همچون روش سنتی آبشاری (conventional waterfall method) ترکیب کرده‌اند.

بنابراین شما چگونه باید تفاوت میان رویکردهای متفاوت مدیریت چابک را درک کنید و تصمیم بگیرید که از کدام فرآیند چابک پیروی کنید؟

من در این مقاله به بررسی چارچوب‌های چابکی می‌پردازم که برای مدیر پروژه‌ای که در دنیای تحول دیجیتال مشغول به کار است می‌توانند مفید واقع شوند. این مقاله به شما کمک می‌کند تا به عنوان یک مدیر پروژه انواع مختلف چابک را درک کنید و نوع روش کاری مناسب‌تری را برای پروژه خود انتخاب کنید. این بدین معنی است که من رویکردهای کمتر شناخته شده و مبهم‌تر را بررسی نمی‌کنم و چارچوب‌های مربوط به مهندسی، پورتفولیو یا چارچوب‌های مربوط به فرهنگ را به طور عمیق بررسی نخواهم کرد.

فهرست متدولوژی‌ها، چارچوب‌ها و رویکردهای چابک

در این مقاله به متدولوژی‌ها، چارچوب‌ها و رویکردهایی اشاره خواهم کرد که همگی ریشه در اصول چابک دارند.



۱. اسکرام (Scrum)
۲. کانبان (Kanban)
۳. اسکرامبان (Scrumban)
۴. دواپس (DevOps)
۵. (باس)دواپس (BUS)DevOps
۶. تفکر طراحی (Design Thinking)
۷. مدیریت پروژه چابک (AgilePM) Agile Project Management
۸. چابک پرنس ۲ (Prince ۲ Agile)
۹. متخصص دارای مدرک (PMI_ACP) (PMI-Agile Certified Professional) PMI-Agile
۱۰. Project Half Double
۱۱. مدیریت برنامه چابک (AgilePgM) Agile Program Management
۱۲. چارچوب چابک مقیاس (SAFe) Scaled Agile Framework
۱۳. اسکرام بزرگ مقیاس (LeSS) Large-Scale Scrum
۱۴. نکسوس (Nexus)
۱۵. اسکرام در مقیاس (S@S) Scrum at Scale
۱۶. مدل اسپاتیفای (Spotify Model)
۱۷. توسعه ناب چابک مقیاس شده (ScALeD) Scaled Agile Lean Development
۱۸. شیفِت چابک یا تغییر جدی چابک (AgileSHIFT)
۱۹. جریان چابک (Agile Fluency)
۲۰. چابکی فضای باز (OSA) Agile Space Agility
۲۱. مقیاس‌های چابک (Agility Scales)
۲۲. هولاکراسی (Holocracy)
۲۳. جامعه‌سالاری (Sociocracy)
۲۴. چابک دیسیپلین‌دار یا چابک اصولی (DA) Disciplined Agile
۲۵. پراکسیس (Praxis)
۲۶. سیستم تولید تویوتا (TPS) Toyota Production System
۲۷. خدمات دیجیتال چابک (AgileDS) Agile Digital Services
۲۸. مدیریت پورتفولیوها (MoP) Management of Portfolios
۲۹. استاندارد مدیریت پورتفولیو (SfPfM) Standard for Portfolio Management
۳۰. مدیریت پورتفولیو چابک (AgilePfM) Agile Portfolio Management
۳۱. مدیریت پورتفولیو مبتنی بر شواهد (E-B PfM) Evidence-Based Portfolio Management
۳۲. مدیریت پورتفولیو دوحالت (Bimoidal PfM) Bimodal Portfolio Management
۳۳. برنامه‌ریزی اکستریم (XP) eXtreme Programming
۳۴. توسعه آزمون‌قبولی محور (ATDD) Acceptance Test Driven Development
۳۵. توسعه آزمون‌محور (TDD) Test-Driven Development



۳۶. توسعه رفتار محور (Behaviour-Driven Development (BDD)
۳۷. توسعه ویژگی محور (Feature-Driven Development (FDD)
۳۸. توسعه آزمایش محور (Experiment-Driven Development (EDD)
۳۹. طراحی تجربه کاربر (User Experience Design (UX Design)
۴۰. تحلیل کسب و کار چابک (Agile Business Analysis (AgileBA)
۴۱. یکپارچه سازی پیوسته / استقرار پیوسته (Continuous Integration/Continuous Deployment (CI/CD)
۴۲. مدلسازی چابک (Agile Modeling (AM)

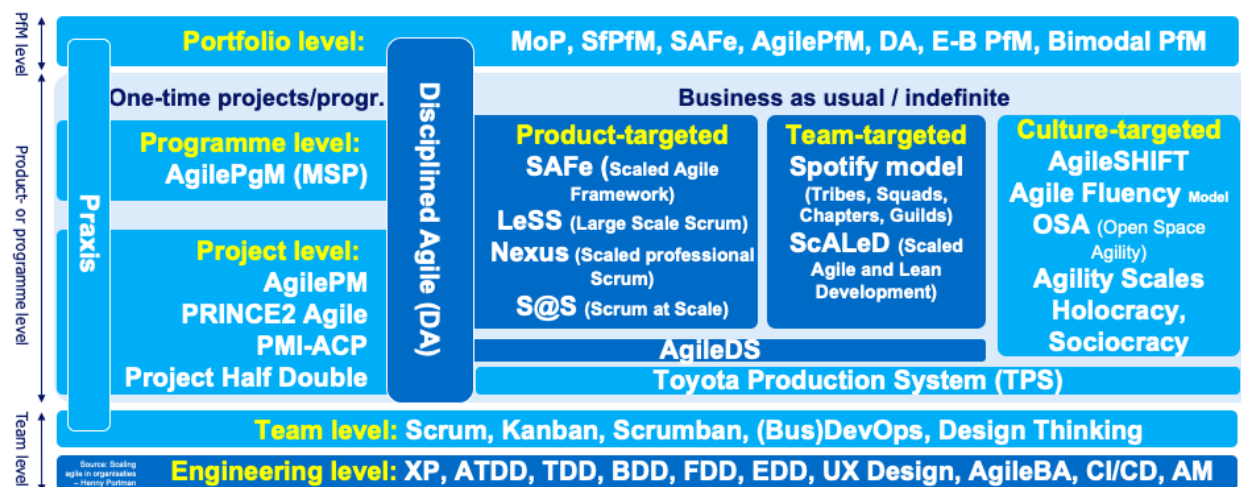
همانطور که گفتیم من به بررسی عمیق رویکردهایی که به سطح پورتفولیو، سطح مهندسی یا فرهنگ سازمانی گرایش دارند نمی پردازم. بنابراین متدولوژی‌هایی را که عمیقا در این مقاله بررسی خواهم کرد را پررنگ کرده‌ام. جزییات متدولوژی‌های دیگر را می‌توانید در وبلاگ من پیدا کنید.

نگاه کلی به جنگل چابک

به منظور اینکه بتوانیم از رویکردهای متفاوت یک برداشت اولیه داشته باشیم، سعی کرده‌ام که یک ساختار برای رویکردها، روش‌ها و چارچوب‌ها در این جنگل ایجاد کنم. در شکل (۱) که آن را «نگاه کلی به جنگ چابک» نامیده‌ام، ۴۴ رویکرد شناخته شده چابک را درون یک ساختار قرار داده‌ام.

این شکل بر اساس یک نسخه ساده‌تر از آن در کتاب مقیاس کردن چابک در سازمان‌ها (Scaling Agile in Organizations) است که آن را در سال ۲۰۱۷ منتشر کرده‌ام. در پاراگراف‌های زیر به توضیح ساختار موجود در شکل می‌پردازم.

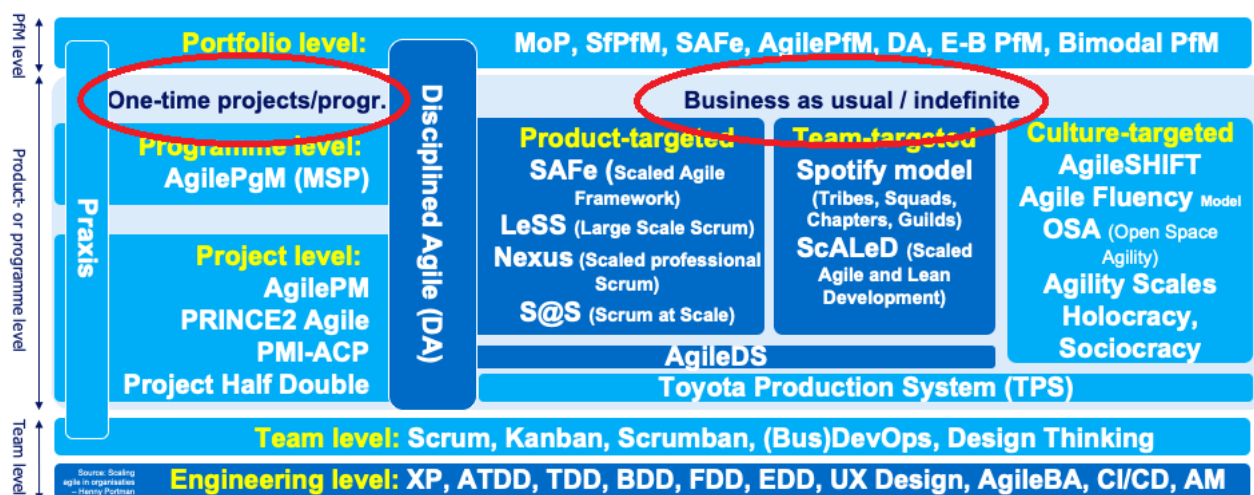
در مستطیل‌های آبی تیره شاهد رویکردهایی هستیم که فقط در سازمان‌های متمرکز بر فناوری اطلاعات وجود دارند. تمام رویکردهای دیگر در مستطیل‌های آبی روشن هستند، بدین معنی که می‌توانند درون سازمان‌های IT محور و غیر IT محور مورد استفاده قرار بگیرند.



رویکردها، چارچوب‌ها و روش‌های کلی چابک



رویکردها، چارچوب‌ها یا روش‌های چابک در دو بخش اصلی قرار می‌گیرند: بخش «برنامه‌ها/ پروژه‌های یکباره» یا بخش «کسب‌وکار معمول/ نامعین». برخی از آنها در هر دو قرار گرفته‌اند بنابراین در کل این بازه گسترده شده‌اند.



من رویکردهای چابک را بر این اساس اینکه در پروژه و برنامه یکباره استفاده می‌شوند یا به طور نامعین در کسب‌وکار معمول استفاده می‌شوند سازماندهی کرده‌ام.

سپس رویکردها، چارچوب‌ها یا روش‌ها بر اساس سطح عملیاتی آنها خوشه‌بندی شده‌اند: سطح مهندسی، سطح تیمی، سطح برنامه یا سطح پورتفولیو. اگرچه سطح تیمی به رنگ آبی روشن است اما هم در توسعه محصول، توسعه خدمت و عملیات IT محور کاربرد دارد و هم در توسعه محصول، توسعه خدمت و عملیات غیر IT محور. در حالی که سطوح مهندسی به صورت ویژه بر توسعه محصول IT محور تمرکز دارند.

من همچنین برخی از روش‌ها را بر اساس هدف آنها خوشه‌بندی کرده‌ام: که آیا معطوف به محصول (product-targeted) هستند یا معطوف به تیم یا معطوف به فرهنگ؟

کسب‌وکار معمولی یا چارچوب‌های چتری نامعین (indefinite umbrella frameworks) که دائماً مورد استفاده قرار می‌گیرند (هم معطوف به محصول و هم معطوف به تیم) به طور ویژه بر روی فناوری اطلاعات و توسعه محصول تمرکز دارند. رویکردهای معطوف به فرهنگ به سازمانها کمک می‌کنند تا چابکی خود را افزایش دهند.

چند اخطار

۱. من در این شکل تمام رویکردها، چارچوب‌ها و روش‌ها را ترسیم نکرده‌ام و اگر بخواهم صادق باشم، بر این باورم که موارد تکراری زیادی وجود دارند و احتمال دارد که پایشان‌های تجاری نیز در آنها نقش داشته باشند و صرفاً به «توسعه» محصول بعدی می‌پردازند که نسبت به رویکردها، چارچوب‌ها و روش‌های موجود ارزش افزوده‌ای ندارد.
۲. مرز محکمی بین کاربردهای فناوری اطلاعات و غیر فناوری اطلاعات وجود ندارد. برای مثال چارچوب‌ها و روش‌های یکباره و موقتی پروژه‌ها و برنامه‌ها هم برای فناوری اطلاعات و هم برای غیر فناوری اطلاعات مفید هستند.



یک نگاه کلی به متدولوژی‌های چابکی که مدیران پروژه دیجیتال باید آنها را بشناسند

در این بخش نگاهی کلی به مهمترین روش‌ها و رویکردهایی می‌اندازیم که مدیران پروژه دیجیتال باید آنها را بشناسند. تمام چارچوب‌ها یا رویکردهای ذکر شده پذیرای بیانیه چابک (Agile Manifesto) هستند و از شکلی از اسکرام استفاده می‌کنند، اما با توجه به عواملی همچون معطوف به تیم یا محصول بودنشان، سطح کاربردشان و عوامل دیگر با هم متفاوت هستند. در مورد اینکه این متدولوژی‌ها چگونه در یک تصویر بزرگ‌تر قرار داده می‌شوند در زیر توضیح داده‌ام.

چابک سطح تیمی

اسکرام (Scrum)

زمانی که تیم‌ها شروع به کار کردن با چابک می‌کنند معمولاً اسکرام را انتخاب می‌کنند. این انتخابی مشخص است، اما سوال این است که آیا اسکرام همیشه گزینه‌ی درست است یا خیر. در پست بلاگ رومن پیچلر این ارتباط با مرحله‌ی چرخه عمر محصول ایجاد شده است.

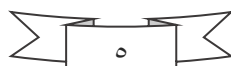
اسکرام (قرارگرفته در سطح تیمی در شکل ۱) توسط کن شوابر (Ken Schwaber) و جف سوترلند (Jeff Sutherland) در راهنمای اسکرام آنها تعریف شده است.

در طی اولین مرحله از چرخه عمر یک محصول تجاری، که در آن محصول تجاری در نهایت برای اولین بار به بازار عرضه می‌شود، عدم قطعیت زیاد است و تمرکز بر روی ارائه به موقع اولین محصول آماده برای بازار است. ضرب‌الاجلی تعیین شده است و باید تا آن روز اینکار انجام شود. در این مرحله تمرکز کل تیم بر روی ارائه یک محصول با قابلیت بازاریابی از نظر تجاری است. این توسعه برای اسکرام با توجه به رویکرد تکراری‌اش بی‌نقص است و قادر است تا با عدم قطعیت مواجه شود و به همراه آن بر روی نتیجه کار کند (محصول تجاری). دومین ارائه به بازار به طور اختیاری می‌تواند با یک مجموعه از کارکردهای مهم اتفاق بیافتد که در نهایت یک محصول بالغ را به بازار ارائه کند.

کانبان (Kanban)

علاوه بر اسکرام شما در سطح تیمی شاهد چارچوب‌هایی همانند کانبان (همانطور که در راهنمای کانبان برای تیم‌های اسکرام توصیف شده است) خواهید بود یا وابستگان آن همانند اسکرامبان، دواپس و (باس) دواپس. سطح تیمی می‌تواند هم در درون محیط IT و هم در محیط غیر IT باشد.

در طی مسیر بعدی چرخه عمر محصول شاهد کاهش میزان عدم قطعیت و تغییرات مورد نیاز خواهیم بود. در این زمان شما می‌توانید به خوبی از کانبان استفاده کنید. در یک جریان پیوسته، داستان‌های کاربران می‌توانند جمع‌آوری شوند و توسعه یافته و یک به یک توسط اعضای مجزای تیم مستقر شوند.



در صورتی که فقط یک تیم چابک برای توسعه و نگهداری از یک محصول یا خدمت وجود داشته باشد و این تیم از اسکرام یا کانبان استفاده کند و یک مالک محصول که بک‌لاگ (backlog) در اولویتش است وجود داشته باشد، در این صورت می‌توانید بپرسید که آیا نیازی به یک مدیر پروژه است یا نه؟ من می‌گویم که نیازی نیست، این تیم را به حال خودشان بگذارید.

دواپس (DevOps)، باس دواپس ((Bus(DevOps) و CI/CD

اگر کسی به جابجایی معمولاً دشوار در محیط‌های تولیدی نگاه کند، می‌بیند که زمان ارائه به بازار را می‌توان با برنامه‌ریزی صحیح این جابجایی و کاهش تعداد خطاهای این جابجایی در زمان ادغام تیم‌های تولید و توسعه و با خودکارسازی آزمون یکپارچگی استقرار کاهش داد (یکپارچگی پیوسته و استقرار پیوسته). CI/Cd بدین صورت یک تیم دواپس ایجاد خواهد شد.

اسکرامبان (Scrumban)

اسکرامبان ترکیبی از اسکرام و کانبان است. در درجه اول به عنوان یک مدل جابجایی برای تغییر از اسکرام به کانبان در نظر گرفته شد و به تیم اجازه داد تا مفاهیم ناب و کانبان را تجربه کنند. امروزه اسکرامبان رویکردی است که در آن تیم انتخاب کرده است تا با توجه به اسکرام و فواصل زمانی (sprints) کار کند، اما از سیستم کانبان برای مشاهده پیوسته و بهبود روش کاری به منظور بهینه‌سازی جریان واحدهای کار (مثلاً داستان‌های کاربران) استفاده کند.

تغییر مقیاس به سمت چابک سطح محصول یا چابک سطح برنامه (Product or Program-Level Agile)

برای اینکه بتوان از یک روش چابک کاری در یک سازمان به نسبت بزرگ استفاده کرد، در اغلب سازمانها صرفاً در اختیار داشتن تیم‌های چابک مجزا کافی نخواهد بود. مثال‌هایی از سازمانهایی وجود دارد که در حال ایجاد یک معماری با اتصال ضعیف-loosely (coupled architecture) بر اساس میکروسرویس‌ها هستند (مثلاً سایت Bol.com در کشور هلند). هر تیم چابک مستقل یک یا چند میکروسرویس را مدیریت می‌کند.

با این وجود در اغلب سازمانها روش چابک کاری باید تغییر مقیاس داده شود و تا جای ممکن باید به همراستایی‌های اصلی (overarching alignments) پرداخت. اینکار می‌تواند توسط یک مدیر پروژه یا از طریق رسمی کردن این همراستایی انجام شود. زمانی که این همسوسازی (coordination) رسمی شود نقش مدیر پروژه از بین می‌رود اما بسیاری از وظایف مدیر پروژه توسط دیگر افراد به انجام می‌رسند (مثلاً تیم یکپارچه‌سازی در نکسوس (Nexus) و مدیر محصول در SAFe و غیره).

چارچوب‌های متعددی برای رسمی کردن این همسوسازی، مدیریت وابستگی‌ها و یکپارچه‌سازی میان تیم‌های چابک دائمی متفاوت در بخش اجرای کسب‌وکار یا کسب‌وکار معمول وجود دارند که عبارتند از:

نکسوس (Nexus)

نکسوس، همانگونه که در راهنمای نکسوس توصیف شده است چارچوبی برای محصول در حال توسعه یا ابتکارات توسعه نرم‌افزار با سه الی ۹ تیم اسکرام در بازه‌های زمانی ۳۰ روزه است. نکسوس پاسخ کن شوابر (Ken Schwaber)، یکی از بنیانگذاران اسکرام، به مقیاس‌پذیری اسکرام است. به چیزی بیش از اراده و رفتار چابک تیم‌های اسکرام متفاوت نیاز است تا بتوانند با هم کار کنند و یک محصول یکپارچه شده را ارائه دهند. نکسوس بر پایه اسکرام است و بر اساس اسکرام و قوانین و نقش‌های فرمول‌بندی شده در



راهنمای اسکرام است. ما می‌توانیم نکسوس را در سطح تیم و برنامه از چارچوب SAFe قرار دهیم، اما آن چیزی را در سطح پورتفولیو تأمین نمی‌کند.

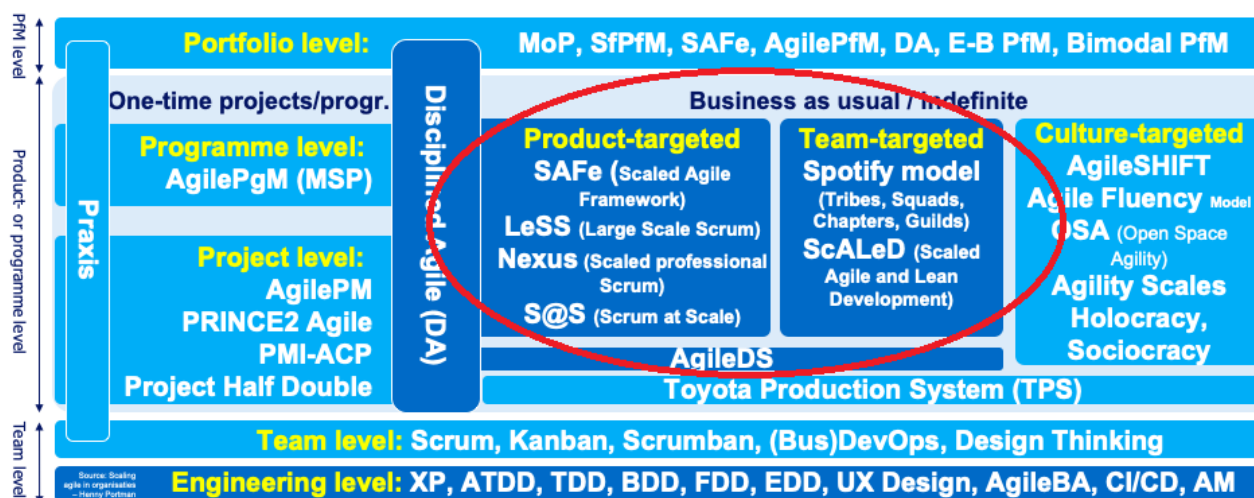
اسکرام در مقیاس (S@S)

اسکرام در مقیاس (S@S) که توسط کریگ لارمن و باس وود توسعه داده شده است یک چارچوب چابک به همراه قوانینی بر اساس اصول و آزمایش‌های عملی است. شرکت LeSS یک پایگاه دانش دسترس‌پذیر رایگان ارائه می‌دهد (LeSS Works) که دربرگیرنده رویکرد، اصول، توصیف فرآیندها، تعریف‌ها، نقش‌ها، مثال‌ها و غیره برای توسعه محصول در مقیاس بزرگ به ویژه مربوط به IT است. شفافیت نیز یک مفهوم کلیدی در درون LeSS است. اولین نسخه آن از سال ۲۰۰۵ است و از آن زمان کار به طور پیوسته بر روی کاربرد و توسعه فراتر LeSS بوده است.

چارچوب چابک مقیاس (SAFe)

چارچوب چابک مقیاس (SAFe)، که توسط دین لیفینگول (Dean Leaffingwell) توسعه یافته است، چارچوبی برای توانمندسازی تغییر مقیاس تیم‌های چابک به منظور ایجاد سیستم‌های بهتر، درگیرسازی بیشتر کارمندان و بکارگیری ملاحظات هزینه‌ای صحیح است. این رسالت یک سازمان چابک مقیاس و بنیانگذار SAFe، دین لیفینگول (Dean Leaffingwell) است. سازمان چابک مقیاس یک پایگاه دانش ارائه می‌دهد که برای دسترس عموم آزاد است (www.scaledagileframework.com) که دارای یک رویکرد یکپارچه شده به شکل توصیف فرآیندها، تعاریف، مثال‌ها و غیره به منظور توسعه محصول چابک/ناب است.

SAFe بر اساس ۵ شایستگی اصلی است. رهبری چابک-ناب (lean-agile leadership)، چابکی فنی و تیمی، دواپس و آزادسازی نیازها، راهکارهای کسب‌وکار و سیستم‌های ناب و مدیریت پورتفولیو ناب.



SAFe، LeSS، نکسوس و S@S در بخش معطوف به محصول قرار دارند. تمام آنها به مثال‌هایی مربوط هستند که در آن چندین تیم بر روی یک محصول پیچیده واحد یا یک جریان ارزش کار می‌کنند.



اگر دوباره به شکل ۱ در بخش کسب و کار معمول/نامعین نگاه کنیم، می‌توانیم روش‌های ذکر شده در بالا یعنی Safe ، LeSS ، نکسوس و S@S را در بخش معطوف به محصول ببینیم. تمام آنها به مثال‌هایی مربوط هستند که در آنها تیم‌های متعدد بر روی یک محصول پیچیده واحد یا یک جریان ارزش کار می‌کنند (چارچوب‌های معطوف به محصول).

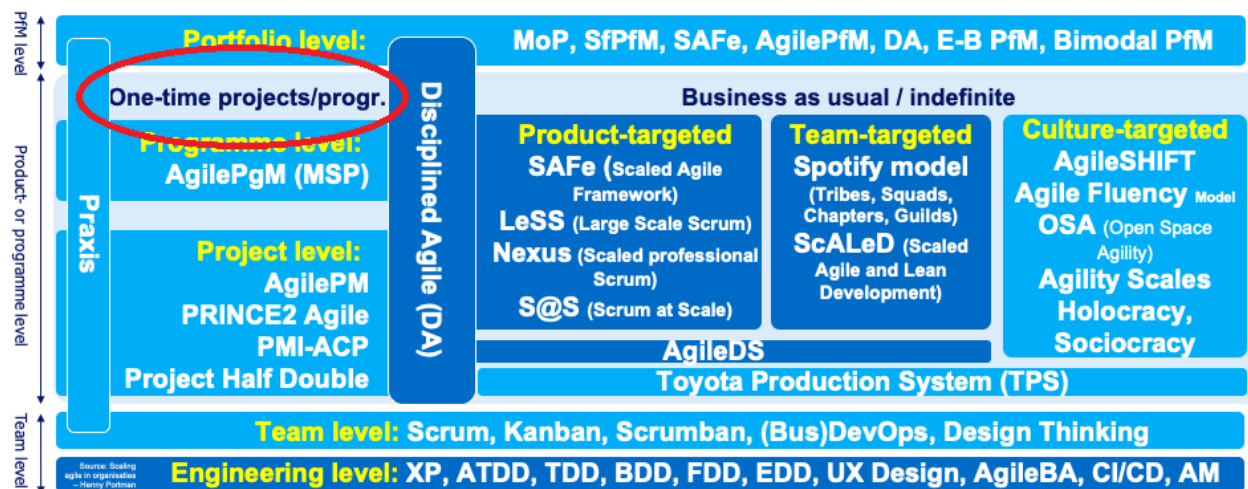
بین محصول و اهداف تیمی یک تقسیم‌بندی وجود دارد که بر اساس همکاری بین تیم‌ها است. به عبارت دیگر آیا تیم‌های مجزا قادرند به طور مستقل کار کنند (با یک تمرکز بر روی تیم) یا باید با یکدیگر برای ارائه‌ی یک محصول جدید یا اصلاح شده کار کنند (با یک تمرکز بر روی محصول).

علاوه بر این چندین رویکرد (که در شکل فهرست نشده‌اند) بین محصولاتی که نیاز به همکاری بین ماکزیمم ۹ تیم (در کل تیم تیم‌ها نباید از عدد دانبار ۱۵۰-۱۲۵ نفر فراتر رود) و یک تیم از تیم‌های تیم‌ها (مثلا راهکارهای بزرگ Safe، نکسوس پلاس، LeSS Huge) تمایز ایجاد می‌کنند. مثالی از این مورد در دنیای واقعی در شرکت فیلپس اتفاق افتاده است که یک تامین کننده تجهیزات پزشکی آلمانی است که از Safe استفاده می‌کند و با ۳۰-۲۰ تیم بر روی یک محصول واحد کار می‌کنند.

مدل اسپاتیفای (Spotify) و ScaLeD

با نگاه به چارچوب‌های معطوف به محصول می‌توانیم شاهد رویکردهایی باشیم که از دپارتمان‌های فناوری اطلاعات برای حفظ و نگهداری دوازده‌ها یا صدها اپلیکیشن و خدمت پشتیبانی می‌کنند. در این تنظیمات وابستگی میان تیم‌ها حداقل است (چارچوب‌های معطوف به تیم‌های متعدد).

در اینجا مدل اسپاتیفای (که توسط هنریک نیبرگ (Henrik Kniberg) ، اندرز ایوارسون (Anders Ivarsson) و خواکیم ساندن (Joakim Sundén) توسعه یافته است) می‌تواند جانمایی شود، اما توسعه ناب چابک مقیاس شده (ScaLeD) که توسط پیتر یک، مارکوس گارتنر، کریستوفر متیس، استفان روک و اندریاس شلیپ توسعه یافته است) نیز می‌تواند مورد استفاده قرار بگیرد. برای هر دو گروه اینترفیس‌های اساسی بین تیم‌ها در حوزه‌هایی همچون یکپارچگی داده‌ها، امنیت و معماری وجود دارد که ممکن است در هنگام پیاده‌سازی تغییرات نیاز به همسوسازی داشته باشند. من انتظار ندارم که در این گروه یک مدیر پروژه پیدا کنم. زیرا این تیم‌ها مستقل هستند و بر این باورم که یک مدیر پروژه ارزش افزوده‌ای ایجاد نمی‌کند.



پروژه‌های یکباره پروژه‌هایی هستند که در آنها بیش از هر جا به یک مدیر پروژه نیاز است.



در سمت چپ شکل ۱ می‌توانیم چارچوب‌های پروژه‌های یکبار را ببینیم. اینها چارچوب‌هایی هستند که در آنها نقش مدیر پروژه مورد نیاز است. اغلب آنها توسعه‌ی چارچوب‌های مدیریت پروژه سنتی هستند:

مدیریت پروژه چابک (AgilePM)

مدیریت پروژه چابک AgilePM که از DSDM حاصل شده است از کنسرسیوم کسب‌وکار چابک (Agile Business Consortium) نشأت می‌گیرد. این روشی است که توسط آن احتمالاً تیم‌های چابک دائمی و تیم‌های غیر چابک در زمان انجام یک پروژه با یکدیگر همسو می‌شوند.

روش AgilePM برای مدیریت کردن پروژه‌های چابک از چارچوبی تشکیل شده است که ترکیبی از فلسفه، اصول حاصل و چهار بلوک سازنده‌ی افراد، فرآیندها، محصولات و کاربردها است.

فلسفه‌ی AgilePM این است که هر پروژه‌ای باید با اهداف معین و آشکار کسب‌وکار همراستا باشد و باید بر روی ارائه زودهنگام محصولاتی تمرکز کند که فراهم‌کننده ارزش افزوده واقعی برای سازمان تجاری باشند.

چابک پرنس ۲ (Prince ۲ Agile)

چابک پرنس ۲ (که از پرنس ۲ از AXELOS حاصل شده است) شامل هر دو پرنس ۲ موجود به عنوان روش تفکر چابک است و روش تفکر چابک باید به صورت رفتار، مفاهیم، چارچوب‌ها، نواحی تمرکز و تکنیک‌های چابک در نظر گرفته شود. اصول، فرآیندها و زمینه‌های موجود در پرنس ۲ باقی می‌مانند اما باید با استفاده از روش چابک کار کردن و خود پروژه شخصی‌سازی شوند.

چابک پرنس ۲ به دنبال بهترین در هر دوی این دنیاهاست که در آنها تاکید بر روی استفاده از پرنس ۲ در درون جهت‌گیری پروژه و مدیریت پروژه و یک رویکرد چابک در ارائه محصول است. بسته به وضعیت پروژه شما می‌توانید از پرنس ۲ یا روش چابک تفکر کمابیش استفاده کنید.

چابک PMI-ACP

فعال دارنده مدرک PMI-Agile یا PMI-ACP (علاوه بر راهنمای PMBok برای PMI) یک چارچوب تنها نیست، بلکه یک مدرک براساس کتاب‌های مختلف (و چارچوب‌ها و تکنیک‌های پایه‌ی توصیف شده در آن) است.

درون PMI-ACP هفت حوزه شناسایی شده اند که هر کدام از آنها به چند زیرحوزه وظیفه‌ای تقسیم شده‌اند. این حوزه‌ها عبارتند از اصول و چارچوب چابک، ارائه ارزش محور، مدیریت ذینفعان، عملکرد تیمی، برنامه‌ریزی سازگار شونده، شناسایی مشکل و راهکار و بهبود پیوسته.

Project Half Double

Project Half Double توسط اجتماعی از فعالان مدیریت پروژه اداره می‌شود که در رابطه با کار خود بسیار مشتاق هستند. این متدولوژی به روشی تکراری و توسط اجتماعی از فعالان مدیریت پروژه هم‌خلق شده است.



این متدولوژی بر اساس چهار بلوک سازنده برای دستیابی به تأثیرات دوبرابر در نصف زمان است: تاثیر، جریان، رهبری و جابجایی محلی. کل این تأثیرات مرتبط با رضایت دینفعان است. جریان نمایش دهنده شدت زیاد و تعاملات مکرر در کار پروژه، یادگیری و تأثیر است. رهبری نشان می‌دهد که شما به عنوان یک رهبر باید پذیرای عدم قطعیت باشید و پروژه را به پیش ببرید. جابجایی محلی این حقیقت است که شما باید متدولوژی را برای نیازهای یک سازمان شخصی‌سازی کنید.

چابک اصولی (Disciplined Agile)

چابک اصولی هم پروژه‌ها و برنامه‌های یکباره و هم توسعه محصول برای کسب و کار معمول را پوشش می‌دهد. جعبه ابزار DA یک جعبه ابزار تصمیم‌گیری فرآیندی است که نحوه توسعه نرم افزار چابک، دواپس، فناوری اطلاعات و کار تیم‌های کسب و کار را در تنظیمات سطح سازمان توصیف می‌کند.

چابک اصولی به ارائه یک پورتفولیو می‌پردازد که در آن علاوه بر پروژه‌ها تعدادی از ابعاد کسب و کار معمول وجود دارند که شامل تیم‌های دائمی و مدیریت عملیات راهکارهای فناوری اطلاعات فعلی می‌شود.

خدمات دیجیتال چابک (Agile Digital Services (AgileDS)

خدمات دیجیتال چابک برای ارائه و عملیات پشتیبانی و نگهداری از آن خدمات (تیم ارائه چابک دائم با استفاده از چرخه عمر محصول/خدمت) وجود دارد.

در مورد متدولوژی‌ها، چارچوب‌ها و رویکردهای چابک بیشتر یاد بگیرید

تا به حال بیش از ۵۰ رویکرد، چارچوب و روش وجود داشته‌اند و هنوز هم این تعداد در حال افزایش هستند. شکل موجود در این نوشته می‌تواند به ما در فرآیند انتخاب رویکرد چابکتان کمک کند. شما می‌توانید بسیاری از توضیحات بیشتر در مورد متدولوژی‌های چابک به ویژه متدولوژی‌های سطح پورتفولیوی مربوط به مهندسی را پیدا کنید که من در اینجا در وبلاگ عمیقاً در موردشان صحبت نکرده‌ام.

با این وجود هرچه بگویم کافی نیست: متعصبانه رفتار نکنید، یک رویکرد، چارچوب یا روش را به صورت یک نوش دارو نبینید که می‌تواند ناگهانی پیاده‌سازی شود.

عقل سلیم نیز در دستیابی به چابکی بسیار بیشتر کمک می‌کند. در واقع بهترین مسیر برای بیشتر چابک شدن، بکارگیری یک رویکرد ساده برای تقسیم‌بندی محصولات و خدماتتان به بخش‌های مستقل کوچکتر و پشتیبانی از آنها توسط یک تیم مستقل است.

منبع: thedigitalprojectrmanager.com

