[illegible]

START
Initiate Project
Define Requirements
Development 1
Development 2
Development n
RELEASE
Accept ?
YES
NO
Deploy To Customer
Implement Changes
Adjust, Track Re-prioritize
Next Iteration Into Development
AGILE
METHODOLOGY

[illegible]

[illegible] **207** [illegible] «[illegible]»[illegible]

اثرگذاری یک محصول بر فعالیت کارکنان و کل مجموعه را پیش از بهکارگیری عملی آن ارزیابی کرد . به عبارت دقیقتر مشتریانی که سفارش ساخت محصولی را میدهند بهدرستی نمیدانند نرمافزار جدیدی که سفارش دادهاند چگونه با سیستمهای دیگر ارتباط برقرار خواهد کرد و کدامیک از فعالیتهای کارمندان آنها بهطور کامل تحت تاثیر استقرار نرمافزار جدید قرار خواهد گرفت . یک محصول نرمافزاری تنها زمانیکه بهطور کامل استقرار پیدا کرد و کارکنان با آن کار کردند، کاستیهای خود را نشان خواهد داد . طراحی و استقرار یک سایت نمونه روشنی در این زمینه است .

نکته مهمی که شرکتهای نرمافزاری باید به آن توجه داشته باشند، بهکارگیری مدل درستی برای توسعه نرمافزار است . شما در نخستین برخورد با مشتری و درخواستی که ارائه میکند باید بهسرعت متوجه شوید که چه مدلی برای ساخت محصول مناسب است. همواره باید به این اصل مهم توجه داشته باشید که مشتریان شما ممکن است پس از عقد قرارداد و در زمان طراحی یا پیادهسازی نرمافزار درخواست خود را تغییر داده یا درخواست جدیدی ارائه کنند، در چنین شرایطی سیستم نیازمند کار و آزمایش مجدد است . در اینگونه موارد بهکارگیری یک مدل آبشاری معمولی یا فرآیندی که بر اساس خصوصیات پایه و ثابت کار میکند، در عمل زمان طراحی محصول را طولانیتر کرده و ممکن است زمان تحویل محصول را به تاخیر اندازند .

انتخاب متدولوژی درست کلید موفقیت کسبوکار شما

شرکتها و بهویژه استارتآپها در فضایی کار خود را آغاز میکنند که سرعت حرف اول را میزند . این موضوع از دو منظر قابل بررسی است . موردی که در پاراگراف قبل به آن اشاره شد یکی از مشکلات رایج شرکتهای نرمافزاری است و متاسفانه اگر شرکتی به این درخواستهای در حال تغییر مشتری بهدرستی پاسخ ندهد یا بیتوجهی کند، در کوتاهترین زمان مشتری خود را از دست خواهد داد . به دلیل اینکه تکنیک بازاریابی دهانبهدهان (یک الگوی موثر در بازاریابی استارتآپی) به روش معکوس کار کرده و اعتبار شرکت را خراب خواهد کرد .

اما نکته بسیار مهم دیگری که باید به آن توجه داشته باشید، بهویژه اگر مالک یک کسبوکار نوپا هستید در ارتباط با تحویل بهموقع محصول است . مهمترین عاملی که باعث شکست استارتآپها و شرکتهای نرمافزار نوپا میشود، تاخیر است . کسبوکار شما محصولی را طراحی کرده، اما درست زمانیکه نرمافزار در دسترس قرار گرفته و آماده استفاده است، دلیل و ایده اصلی که نرمافزار بر اساس آن ساختهشده تغییر پیدا کرده و نرمافزار به محصولی بیمصرف تبدیلشده است . به همین دلیل است که در ارتباط با برخی از کسبوکارهای ویژه ما از فرآیندهای توسعه و تحویل سریع نرمافزار استفاده میکنیم . اگرچه روشهای زیادی برای توسعه سریع نرمافزار وجود دارد اما همه آنها دارای یکسری ویژگیهای اساسی هستند . این ویژگیها به شرح زیر هستند :

- طراحی و پیادهسازی همزمان با هم انجام میشود . در این مدلها هیچگونه جزئیاتی در ارتباط با ویژگیهای سیستم وجود نداشته و طراحی مستندات به شکل حداقلی است. مستنداتی که بهطور عمده در محیط برنامهنویسی نوشته میشوند .
- سیستم در وضعیت توسعه افزایشی طراحی میشود. کاربران نهایی و دیگر ذینفعان سیستم در تعیین و ارزیابی هر یک از این مراحل مشارکت دارند . آنها ممکن است تغییراتی در سیستم به وجود آورده یا نیازمندیهای جدیدی ارائه کنند که باید در مراحل بعدی توسعه اعمال شوند .
- رابطهای کاربری سیستم در بیشتر موارد با استفاده از یک سیستم توسعه تکاملی که اجازه میدهد، طراحی رابط بهسرعت با استفاده از آیکونها و تصاویر ساخته شود، انجام گیرد . سیستم در ادامه ممکن است یک رابط وبمحور را برای یک مرورگر یا یک رابط برای یک پلتفرم خاص همچون ویندوز مایکروسافت تولید کند .

متدولوژیهای توسعه افزایشی و سریع که در شمارههای گذشته مجله (193 و 189) به آنها اشاره داشتیم، شامل تولید و تحویل نرمافزار در حالت افزایشی بهجای یک محصول واحد است. هر فرآیند تکرارشونده یک محصول افزایشی جدیدی را تولید میکند . مدل افزایشی یک مزیت اصلی در زمان توسعه نرمافزار به همراه دارد و آن چیزی نیست جزء تحویل سریع محصول به مشتری.

مطلب پیشنهادی

[illegible]

[illegible]

[illegible] (1)

Define system deliverables
Design system architecture
Specify system increment
Build system increment
Validate increment
Integrate increment
Validate system
System complete?
NO
YES
Deliver final system

[illegible]

[illegible]

[illegible]

است؟

متدولوژیهای چابک

در دهه 80 و اوایل دهه 90 میلادی، دیدگاهی با این مضمون رواج پیدا کرد که بهترین راه برای ساخت نرمافزارهایی با بالاترین کیفیت، بهکارگیری روشهایی است که از ابزارهای CASE پشتیبانی میکنند . ابزارهایی که به تیمهای توسعه اجازه میدهند بر فرآیندهای توسعه نرمافزار کنترل دقیقی داشته باشند . این دیدگاه بهویژه از طرف جامعه مهندسانی ارائه شد که در زمینه طراحی و ساخت پروژههای بزرگ به فعالیت اشتغال داشتند . افرادی که وظیفه آنها ساخت سامانههایی با طول عمر بالا بود که از ترکیب تعداد زیادی سامانه منفرد از یکدیگر به وجود میآمدند . در سال 2001 انجمنی متشکل از توسعهدهندگان نرمافزار و طرفداران روشهای متدولوژیهای سبکوزن به وجود آمد که سعی داشتند ویژگیهای مختلف و شاخص مبتنی بر متدولوژیهای سبکوزن را شناسایی کنند . این انجمن با هدف رسیدن به دیدگاهی منسجم و واحد گرد یکدیگر آمدند و سرانجام موفق شدند بیانیه مشترکی را که مانیفست (Manifesto) نامیده میشود، منتشر کنند . در این بیانیه یکسری اصول، ارزشها و قوانین برای توسعه سریع نرمافزارها معرفی شدند که به اصول متدولوژی چابک معروف شدند. از مهمترین اصولی که این بیانیه مشترک بر آنها تاکید دارد، موارد زیر میتوان اشاره کرد :

- فراهم آوردن زمینه رضایت مشتری از طریق تحویل سریع و مداوم نرمافزار
- انعطافپذیر بودن در برابر تغییرات حتی اگر پروژهها در مراحل پایانی طراحی قرار داشته باشند
- انتشار نسخههای آزمایشی نرمافزار در بازههای زمانی منظم
- برقراری ارتباط منسجم میان کاربران و تیم توسعهدهنده نرمافزار
- مستندسازی قابل استناد
- پشتیبانی از یک فرآیند تولید یکنواخت و باثبات
- بهکارگیری افراد متخصص و واجد شرایط
- سادهسازی کارها با حذف موارد غیرضروری
- بررسی عملکرد و کارایی تیم در بازههای زمانی مشخص

بهطورکلی، مدل توسعه نرمافزاری چابک را مجموعهای از متدولوژیهای توسعه نرمافزار شکل میدهند که بر مبنای تکامل تدریجی و تکرارشونده کار میکنند . مجموعهای که در آن، راهحلها از طریق خودسازماندهی و همکاری فیمابینی تیمها به دست میآید . به عبارت دقیقتر، متدولوژی چابک یک چارچوب مفهومی است که سعی میکند تعامل میان مشتریان و تیم توسعهدهنده در سراسر چرخه توسعه حفظ شود. از مهمترین مدلهای مبتنی بر متدولوژی چابک میتوان به مدل افزایشی (Incremental)، تعاونی (Cooperative)، مستقیم (Straightforward)، تطبیقی (Adaptive) و.... اشاره کرد . شکل (2) مدل توسعه چابک را نشان میدهد .

Develop Functionality 1
Integrate and Test
Develop Functionality 2
Integrate and Test
Develop Functionality N
Integrate and Test
Demo Release
Client's Feedback
Make Changes
System Testing
All Functionalities Complete?
Yes
No
Next Iteration
Agile Development

امروزه، مدلهای توسعه سبکوزن مختلفی وجود دارند که هر یک در ارتباط با پروژههای خاصی به کار گرفته میشوند . در ادامه با چند مورد از معروفترین مدلها آشنا خواهید شد .

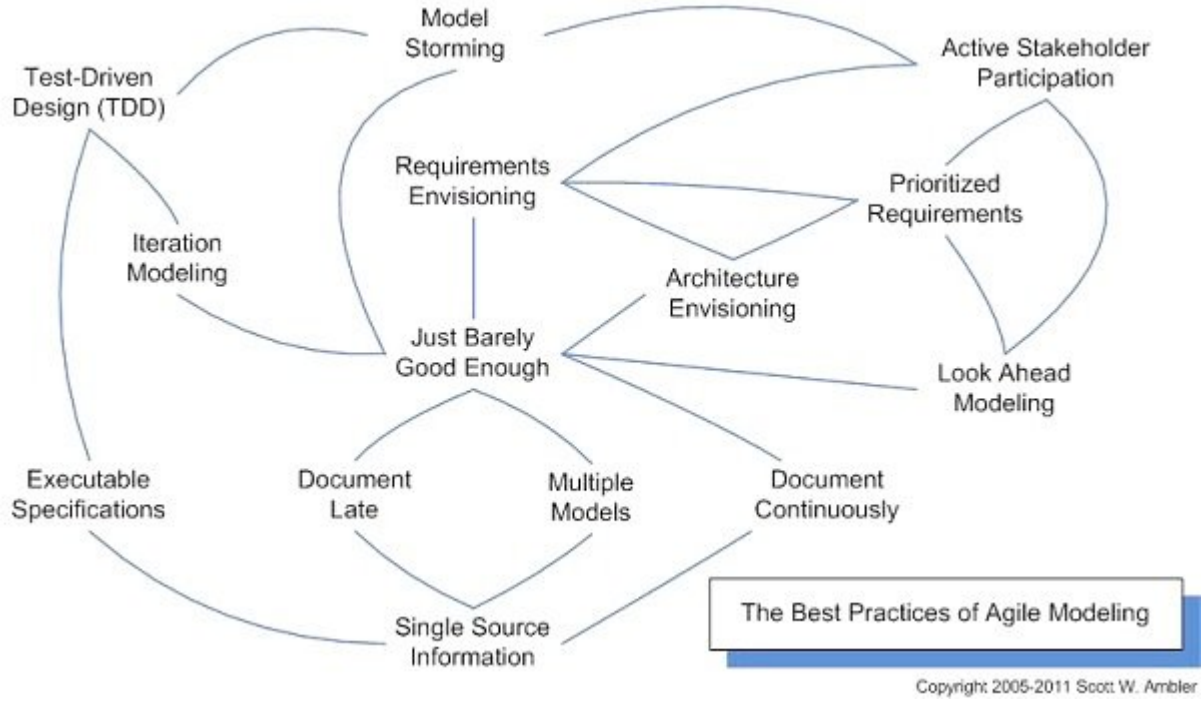
ASD سرنام (Adaptive Software Development)

توسعه تطبیقی نرمافزار، یکی از فرآیندهای توسعه نرمافزار است که توسط جیم های اسمیت و سام بایر طراحی شد. (شکل 3)

Learning Loop
Project Initiation
Adaptive Cycle Planning
Concurrent Component Engineering
Quality Review
Final Q/A and Release
Speculate
Collaborate
Learn

این مدل برای ساخت سیستمها و نرمافزارهای پیچیده استفاده میشود . این مدل بر اساس دو ایده اصلی ماهیت سازگاری مبتنی بر مولفه و همکاری گروهی به کار گرفته میشود . تفکر و تعمق (Speculation) ، همکاری (Collaboration) و یادگیری (Learning) سه فاز اصلی این مدل را شکل میدهند. ASD بهعنوان جایگزینی برای مدل توسعه سنگینوزن آبشاری سنتی ارائه شد . در این مدل از یک الگوی تکرارشوندگی روی فاز اصلی استفاده میشود .

(Agile Modeling) AM



Crystal family

Disciplined agile delivery

Deploy
Initial
Idea
Envision
Hypothesis
Build a
little
Keep going
Observe &
Measure
Proven
Idea
Productize
Pivot
Disproven
Idea
Cancel
Copyright 2014
Disciplined Agile Consortium

[illegible]

[illegible]

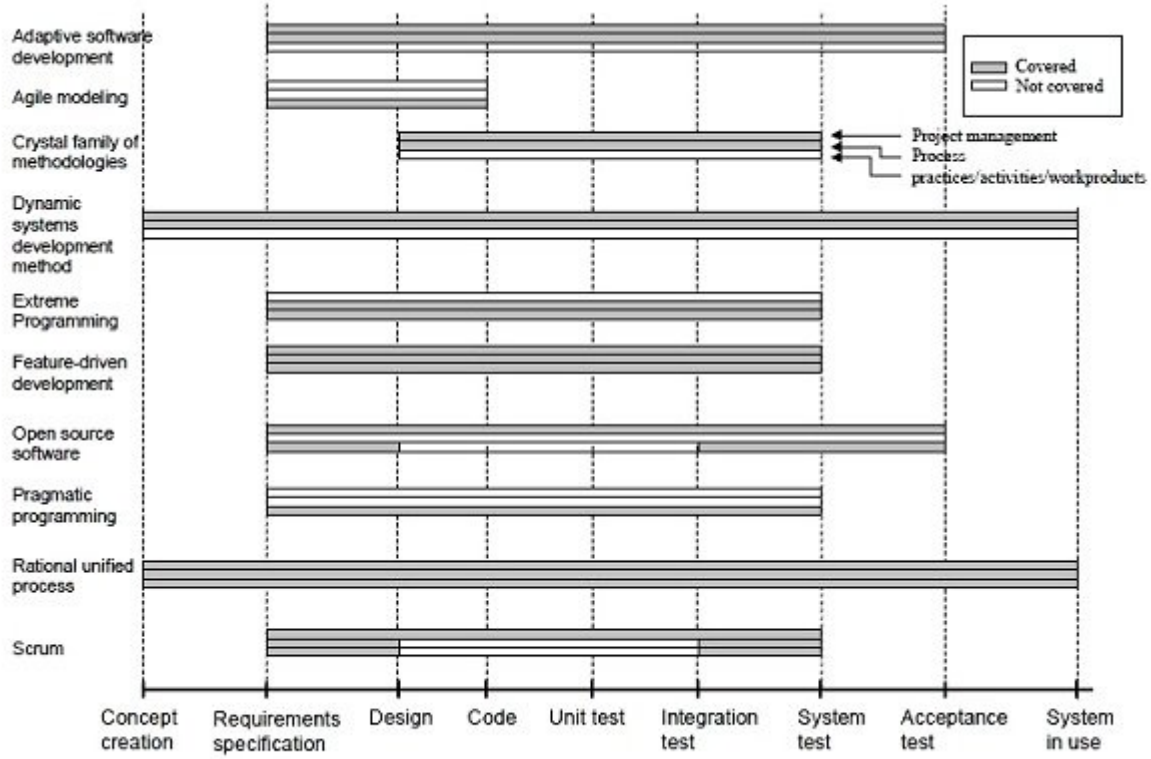
[illegible]

[illegible] V

[illegible]

[illegible]

[illegible] (7 [illegible]).



Adaptive software development
Agile modeling
Crystal family of methodologies
Dynamic systems development method
Extreme Programming
Feature-driven development
Open source software
Pragmatic programming
Rational unified process
Scrum
Covered
Not covered
Project management
Process
practices/activities/workproducts
Concept creation
Requirements specification
Design
Code
Unit test
Integration test
System test
Acceptance test
System in use

بعضی از آنها بر روشها (برنامهنویسی مفرط، برنامهنویسی فعال و مدلسازی چابک)، برخی بر مدیریت پروژههای نرمافزاری (مانند اسکرام)، برخی بر تمام چرخه عمر توسعه نرمافزار (متدهای توسعه پویای یک سیستم و فرآیند توسعه یکپارچه) و برخی دیگر بر تعیین نیازمندیها تاکید دارند . بنابراین، یک تفاوت آشکار بین متدهای گوناگون توسعه چابک نرمافزار وجود دارد . اگرچه متدهای توسعه پویای یک سیستم و فرآیند توسعه یکپارچه نیازی به رویکردهای مکمل برای پشتیبانی از توسعه نرمافزار ندارند، بقیه آنها با درجههای متفاوت به این مکملها نیاز دارند . فراموش نکنید حتی مدلهای پیچیدهای همچون متدهای توسعه پویای یک سیستم میتوانند از سوی هر توسعهدهندهای به کار گرفته شوند .

منبع :
agilemanifesto
disciplinedagiledelivery.wordpress
adfkickstart
en.wikiversity
adaptivesd
versionone
ambysoft
expertiza.ncsu

دسته بندی:
پرونده ویژه
استارتآپ

تاریخ انتشار :
11:10 - 08/09/1397

برچسب:
نرم افزار - توسعه نرم افزار - متدولوژی - متدولوژی های چابک - Agile - اسکرام

نشانی
https://www.shabakeh-mag.com/cover-story/13809/%D9%85%D8%AA%D8%AF%D9%88%D9%84%D9%88%DA%98%DB%8C%E2%80%8C%D9%87%D8%A7%DB%8C-%DA%86%D8%A7%D8%A8%DA%A9-%DA%86%D9%87-

%D9%87%D8%B3%D8%AA%D9%86%D8%AF-%D9%88-%DA%86%D8%B7%D9%88%D8%B1-%D9%81%D8%B1%D8%A2%DB%8C%D9%86%D8%AF-%D8%AA%D9%88%D8%B3%D8%B9%D9%87-%D8%B1%D8%A7-%D8%B3%D8%B1%DB%8C%D8%B9-%D9%88-%D8%B3%D8%A7%D8%AF%D9%87-%D9%85%DB%8C%E2%80%8C%DA%A9%D9%86%D9%86%D8%AF%D8%9F