

دانشگاه فنی و حرفه ای

دانشکده فنی و حرفه ای امام جعفر صادق (ع) آستانه اشرفیه

مهندسی نرم افزار
فصل سوم : توسعه ی چابک
AGILE DEVELOPMENT

مهندسی نرم افزار چابک، تلفیقی از یک فلسفه و مجموعه‌ای از دستورالعمل‌های توسعه است.

این فلسفه مشوق موارد زیر است:

- جلب رضایت مشتری
- تحویل افزایشی نرم افزار از همان ابتدای پروژه.
- تیم های پروژه ای کوچک با انگیزه بالا
- روش های غیر رسمی
- حداقل محصولات مهندسی نرم افزار
- سادگی کلی مهندسی نرم افزار

توسعه چابک

چه کسی این کار را انجام می‌دهد؟ مهندسين نرم‌افزار و طرفهای ذی‌نفع در پروژه با یکدیگر کار می‌کنند و یک تیم چابک را تشکیل می‌دهند.

تیمی که سازمان‌دهی‌اش بر عهده خودش است. سیستم چابک به ارتباطات و همکاری میان همه افراد سیستم رسیدگی می‌کند.

اهمیت؟ مهندسی نرم‌افزار چابک، برای مهندسی سنتی گروه و انواع معینی از پروژه‌های نرم‌افزاری جایگزین منطقی ارائه می‌کند.

مراحل کار: فعالیت‌های چارچوبی پایه (ارتباطات، برنامه‌ریزی، مدل‌سازی، ساخت و استقرار) به قوت خود باقی خواهند ماند، ولی به یک مجموعه وظایف کمینه تغییر شکل می‌دهند.

محصول کار: یک نرم‌افزار عملیاتی است که در تاریخ مناسب به مشتری تحویل شود.

چگونه اطمینان حاصل کنیم که کار، درست انجام شده‌است؟ تولید نرم‌افزار قابل تحویل و رضایت مشتری

مفاد بیانیه توسعه نرم افزار چابک

- افراد و تعامل ها را بر فرآیندها و ابزارها برتری دهیم.
- نرم افزار عملیاتی را بر مستندات جامع برتری دهیم.
- همکاری با مشتری را بر مذاکره و قرارداد برتری دهیم.
- پاسخ به تغییر را بر دنبال کردن یک برنامه برتری دهیم.

روش چابک : فقط برای مهندسی نرم‌افزار نیست بلکه به عنوان فلسفه‌ای جایگزین برای همه کارهای نرم‌افزاری قابل استفاده است.

یکی از بارزترین ویژگی‌های روش چابک : توانایی آن در کاهش دادن هزینه‌های ناشی از تغییر است.

در مدل‌ها معرفی شده در فصل ۲، ضعف و سستی کسانی که نرم‌افزارها را می‌سازند، فراموش می‌شود. مهندسان نرم افزار، روبات نیستند و در سبک کاری باهم تفاوت‌های بزرگ دارند. از جمله :
در سطح مهارت ، خلاقیت، نظم و انضباط ،سرعت عمل.

با اعمال انضباط یا تحمل می‌توان نقاط ضعف افراد را اداره کرد.

5 چون سازگاری درکنش از نقاط ضعف انسان است، روش‌هایی با انضباط بالا شکننده‌اند.

چابکی در حیطة کار مهندسی نرم افزار

هنگام توصیف یک فرآیند نرم افزار مدرن، چابکی واژه‌ای است که به وفور به گوش می‌رسد.

تیم چابک تیمی فرز و چالاک است که قادر است به تغییرات پاسخ مناسب بدهد.

تغییر چیزی است که در توسعه نرم‌افزار، بسیار با آن مواجه می‌شویم.

- تغییرات در نرم‌افزارهایی که در حال ساخته شدن هستند ،

- تغییرات در اعضای تیم ،

- تغییرات به دلیل فناوری جدید ،

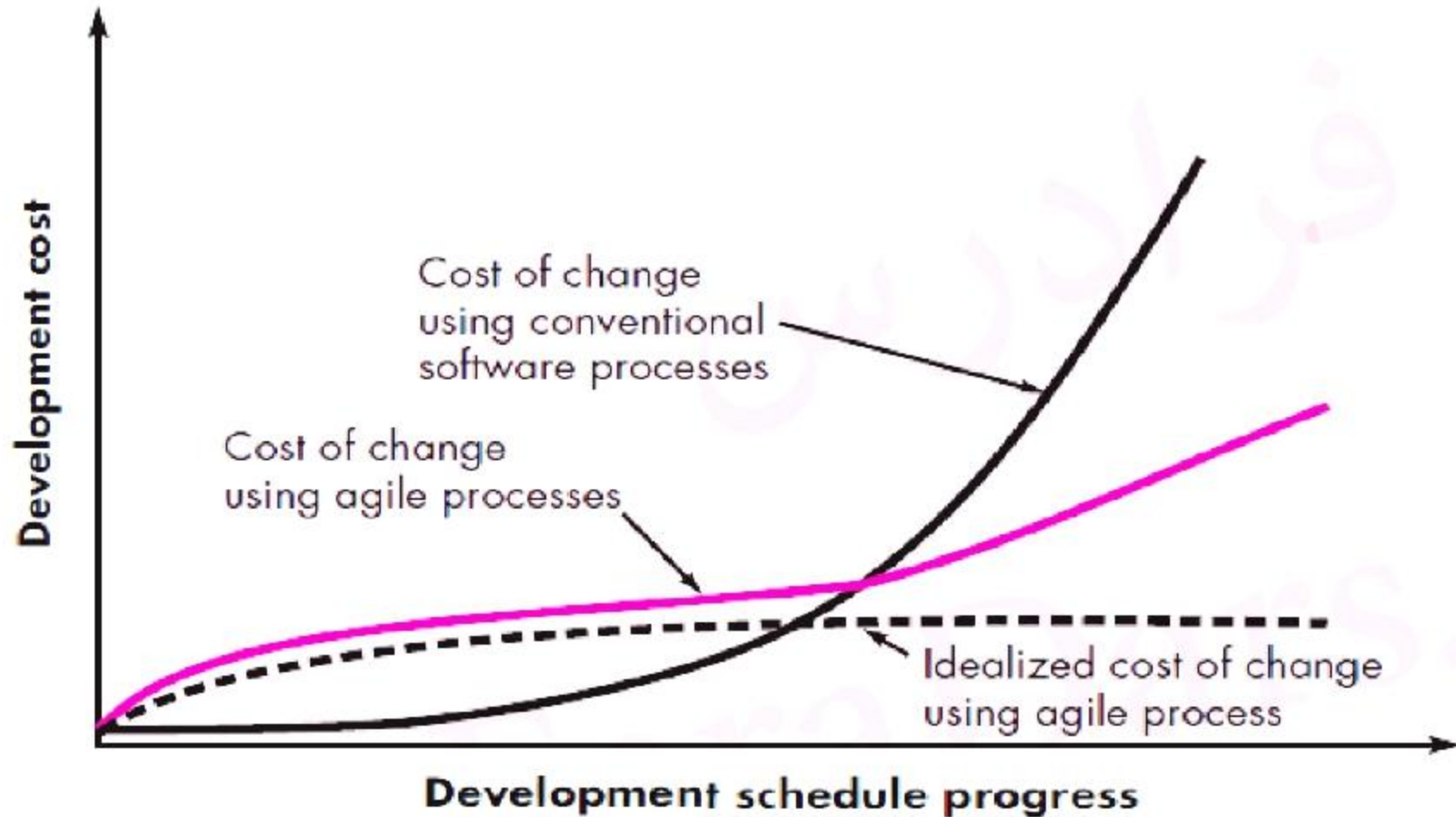
تغییرات از هر نوع ممکن است بر محصول در حال ساخت یا محصول نهایی ، تاثیرگذار باشند.

سیستم چابک می‌داند که نرم‌افزار توسط افرادی توسعه می‌یابد که در قالب تیمی کار می‌کنند و مهارت‌های این افراد و توانایی ایشان در همکاری ، هسته اصلی موفقیت پروژه است.

فراگیر بودن تغییر ، دلیل اصلی برای چابکی است.

مرتکب این اشتباه نشوید که چابکی این مجوز را به شما می‌دهد که بدون برنامه‌ریزی به راهکار برسید. یک فرآیند لازم است و انضباط ضروری است .

چابکی و هزینه‌های تغییر



یک فرآیند چابک با طراحی خوب، منحنی هزینه‌ی تغییر را تسطیح می‌کند و به این ترتیب، سیستم نرم‌افزاری قادر به پاسخ‌گویی به تغییرات در اواخر پروژه خواهد بود، بدون اینکه ضربه‌ای قابل ملاحظه از نظر زمان و هزینه به پروژه وارد آید.

فرآیند چابک هزینه‌ی تغییرات را کاهش می‌دهد چون نرم‌افزار در قالب چند گام روانه می‌شود و تغییرات را در یک گام بهتر می‌شود کنترل کرد.

فرایند چابک چیست ؟

هر فرآیند نرم‌افزار چابک، به گونه ای مشخص می شود که تعدادی از فرض‌های کلیدی را درباره اکثریت پروژه‌های نرم‌افزاری پاسخ گو باشد:

- ۱) پیش‌بینی اینکه کدام خواسته‌های نرم‌افزاری باقی است و کدام یک تغییر می‌کند ، دشوار است.
- ۲) برای بسیاری از انواع نرم‌افزارها ، طراحی و ساخت به طور موازی انجام می‌شود که دشوار است.
- ۳) تحلیل ، طراحی ، ساخت و آزمودن ، ممکن است به آن اندازه که دوست داریم ، قابل پیش‌بینی نباشد.

پرسش :

چگونه فرآیندی ایجاد کنیم که قادر به مدیریت موارد غیر قابل پیش‌بینی باشد؟ فرآیندی با انطباق پذیری بالاتر.
در یک فرآیند چابک باید روند انطباق به طور افزایشی انجام پذیرد. برای دستیابی به آن نیاز به بازخورد از مشتری دارد.

اصول چابکی

دوازده اصل در پیمان چابک :

- ۱- جلب رضایت مشتری با تحویل زود هنگام و پیوسته، بیشترین اولویت را نزد ما دارد.
- ۲- پذیرا بودن تغییرات در خواسته‌ها حتی در اواخر فرآیند توسعه.
- ۳- تحویل پیوسته نرم‌افزارها از دو هفته تا دو ماه .
- ۴- دست‌اندرکاران و افراد تجاری باید در سرتاسر پروژه هرروز با هم کار کنند.
- ۵- سپردن پروژه به افراد با انگیزه، فراهم‌سازی محیط و پشتیبانی مورد نیاز آنها و اطمینان کردن به آنها در انجام کارها.
- ۶- اثر بخش‌ترین و موثرترین روش انتقال اطلاعات به درون و بیرون سیستم توسعه ، گفتگوی رودررو است.

اصول چابکی

- ۷- نرم‌افزاری کاری، میزان اصلی در سنجش پیشرفت است.
- ۸- فرآیندهای چابک، توسعه پایدار را ارتقا می‌بخشد.
- ۹- توجه پیوسته به اعتلای فنی و طراحی خوب، باعث بهبود افزایش چابکی می‌شود.
- ۱۰- سادگی ضروری است.
- ۱۱- بهترین معماری‌ها، خواسته‌ها و طراحی‌ها از تیم‌های خودسازمان‌دهی شده ظهور می‌کنند.
- ۱۲- تیم در بازه‌های منظم، بازخوردی از میزان بهبود اثربخش خود ارائه می‌دهد و سپس رفتار خود را مطابق این بازخورد تنظیم می‌کند.

سیاست توسعه چابک

The Politics of Agile Development

درباره مزایا و قابلیت چابک بحث و جدل فراوان است.

جیم اسمیت می‌گوید: روش‌شناسان، یک مشت آدم‌های فاقد خلاقیت هستند که ترجیح می‌دهند مستندات بدون نقص تهیه کنند تا اینکه سیستمی کاری ارائه دهند که نیازهای تجاری را برآورده کند.

وی می‌گوید: روش‌شناسان چابک یک مشت نفوذگر با استعدادند که وقتی تلاش می‌کنند اسباب‌بازی‌های خود را بزرگ کنند و به نرم‌افزاری در سطح شرکت‌ها تبدیل کنند، کلی ذوق زده می‌شوند.

نکته مهم اینکه با در نظر گرفتن بهترین ایده‌ها از هر دو مکتب، بیشترین بهره‌عاید خواهد شد و چیزی از تخریب دیگری به دست نخواهد آمد.

عوامل انسانی

در این توسعه به عوامل انسانی تاکید فراوان دارند .

توسعه چابک بر استعدادها و مهارت‌های افراد و شکل‌دهی به فرآیند براساس افراد و تیم‌ها موجود تاکید دارد.
فرایند باید بر اساس نیازهای افراد و تیم‌ها شکل پیدا کند نه بر عکس.

خصوصیات زیر باید در میان افراد سیستم و خود سیستم وجود داشته باشد :

۱- رقابت : Competence

شامل استعداد ذاتی، مهارت‌های خاص مرتبط با نرم‌افزار و آگاهی کلی از فرآیندی است که تیم برگزیده است.

۲- کانون توجه مشترک : Common focus

با وجود وظایف هر کدام از اعضای تیم هدف واحد تحویل نسخه جدیدی از نرم‌افزار به مشتری در زمان مقرر.

۳- همکاری : Collaboration

داشتن همکاری در هزینه‌های فرآیند و انتشار اطلاعاتی که برای مشتری ارزش تجاری در بر داشته باشد.

۴- توانایی تصمیم‌گیری : Decision-making ability

آزادی کنترل سرنوشت خود، یعنی اجازه تصمیم‌گیری برای مسائل فنی و پروژه.

۵- توانایی حل مسئله با منطق فازی : Fuzzy problem-solving ability

ممکن مسئله‌ای که امروز در حال حل کردن آن است، فردا مسئله‌ای باشد که دیگر نیازی به حل آن نباشد، ولی درس‌هایی که از حل هر مسئله گرفته می‌شود، ممکن است بعداً در پروژه به کار آید.

۶- احترام و اطمینان متقابل : Mutual trust and respect

باید تیم قوام یافته باشد و اعضای تیم چنان به هم پیوسته باشند که کلیت حاصل چیزی بیش از مجموع اجزای تشکیل‌دهنده باشد.

۷- خودسازمان‌دهی : Self-organization

- تیم چابک خودش را سازمان‌دهی می‌کند تا کارها به انجام برسند .
- تیم فرآیند را سازمان‌دهی می‌کند تا به بهترین نحو در محیط محلی اسکان یابد.
- تیم ، زمان‌بندی را سازمان‌دهی می‌کند.

برنامه‌نویسی حدی

پربکاربردترین رویکرد در توسعه نرم‌افزار به روش چابک است.

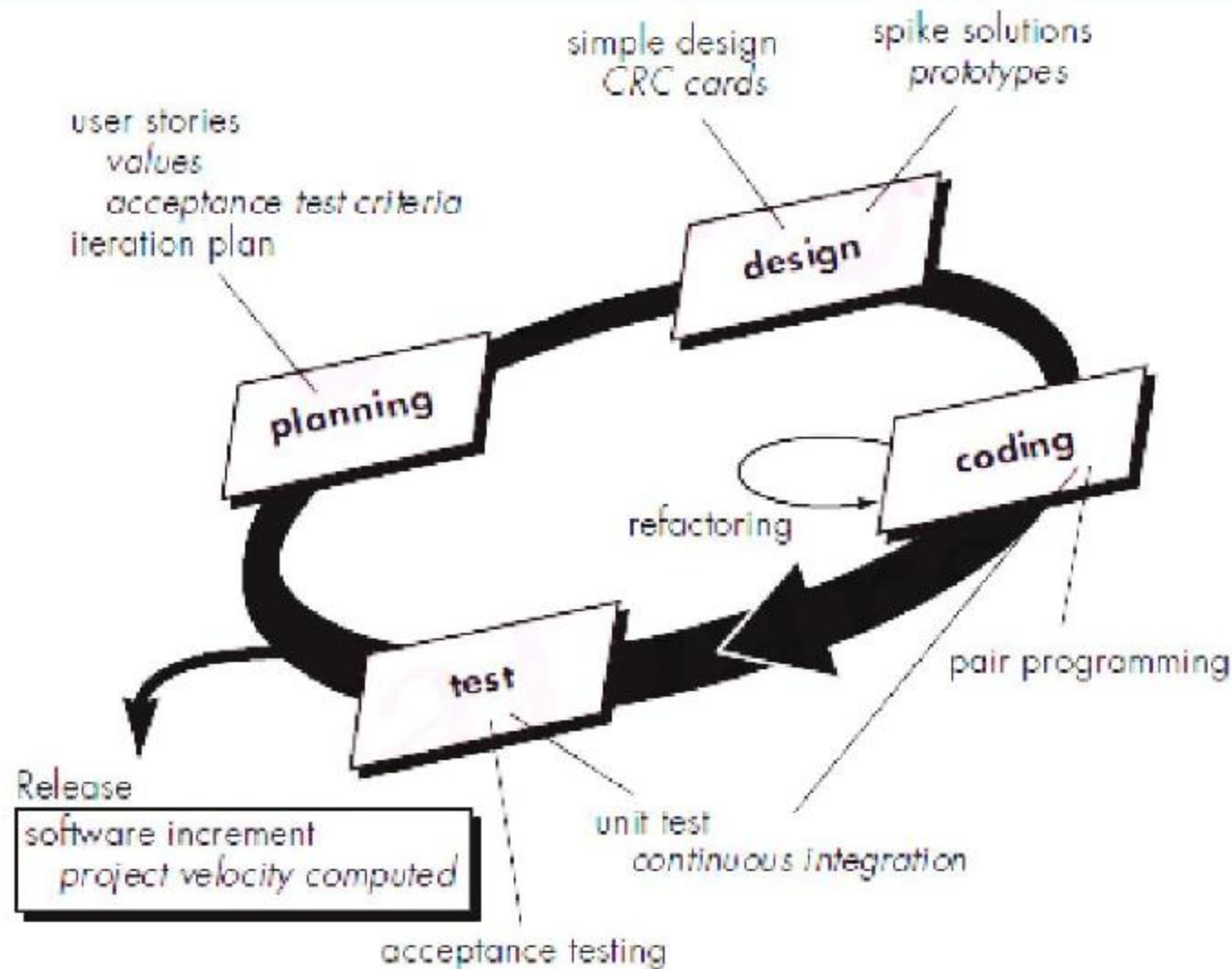
ارزش‌های Xp

- ۱- **ارتباطات**: همکاری نزدیک و در عین حال غیر رسمی میان مشتریان و سازندگان.
- ۲- **سادگی**: ایجاد یک طراحی ساده که به سادگی در قالب کد نویسی قابل پیاده سازی است.
- ۳- **بازخورد**: از سه منبع قابل حصول است: نرم افزار، مشتری، سایر اعضای تیم.
- ۴- **جرات**: تیم xp چابک باید انضباط (جرات) لازم را برای طراحی امروز و خواسته های آینده را داشته باشد.
- ۵- **احترام**: در میان اعضای تیم و سایر طرف های ذی نفع

Xp پاسخ این سؤال است که چقدر می توان
کوچک عمل کرد و هنوز نرم افزارهای بزرگ ساخت.

فرایند xp

این فرایند در حیطه چهار فعالیت چارچوبی رخ می دهد:



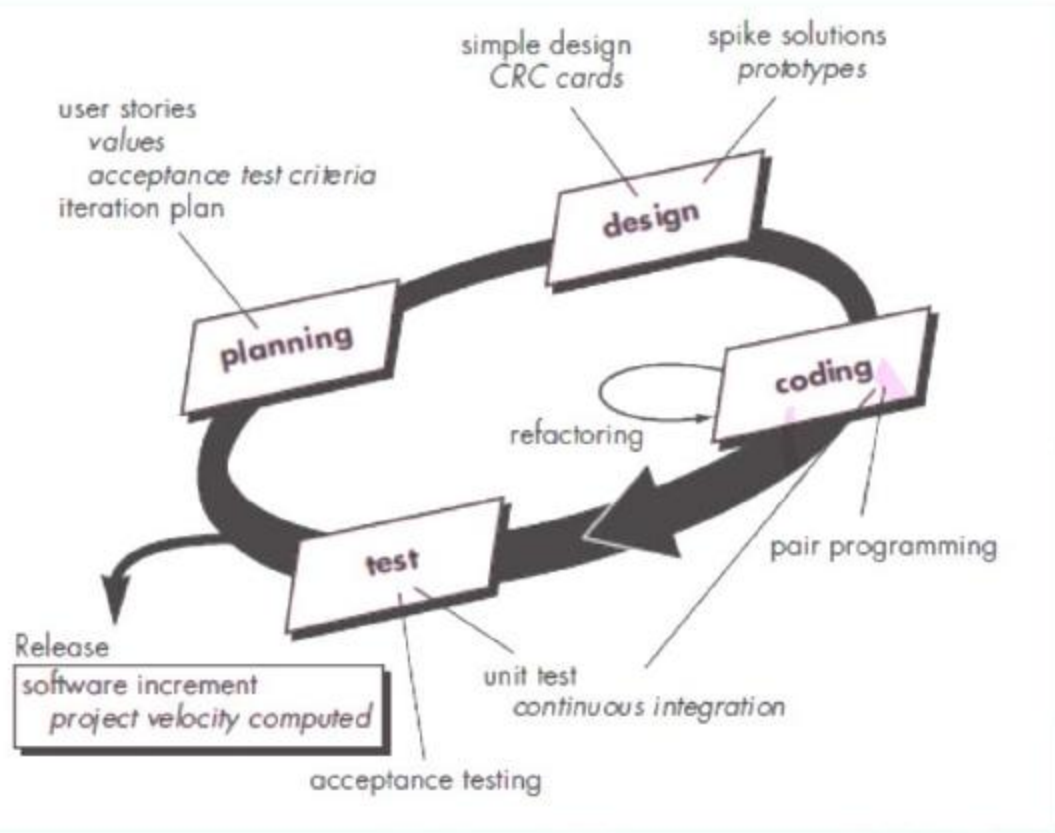
۱- برنامه ریزی

۲- طراحی

۳- کد نویسی

۴- آزمون

فرایند xp (برنامه ریزی)



۱- گوش کردن و جمع آوری خواسته های مشتری و شناخت حیطه تجاری مناسب

۲- تعیین خروجی ها از روی "داستان ها"

۳- اولویت بندی یک خروجی به خروجی دیگر (خروجی ها و برنامه ها) توسط مشتری

۴- گروه بندی اولویت ها با همکاری مشتریان و سازندگان

چگونگی گروه بندی و ترتیب داستان ها:

۱- همه نیازها بلافاصله پیاده سازی شود(در عرض چند هفته)

۲- اولویت های بالاتر زودتر انجام شود

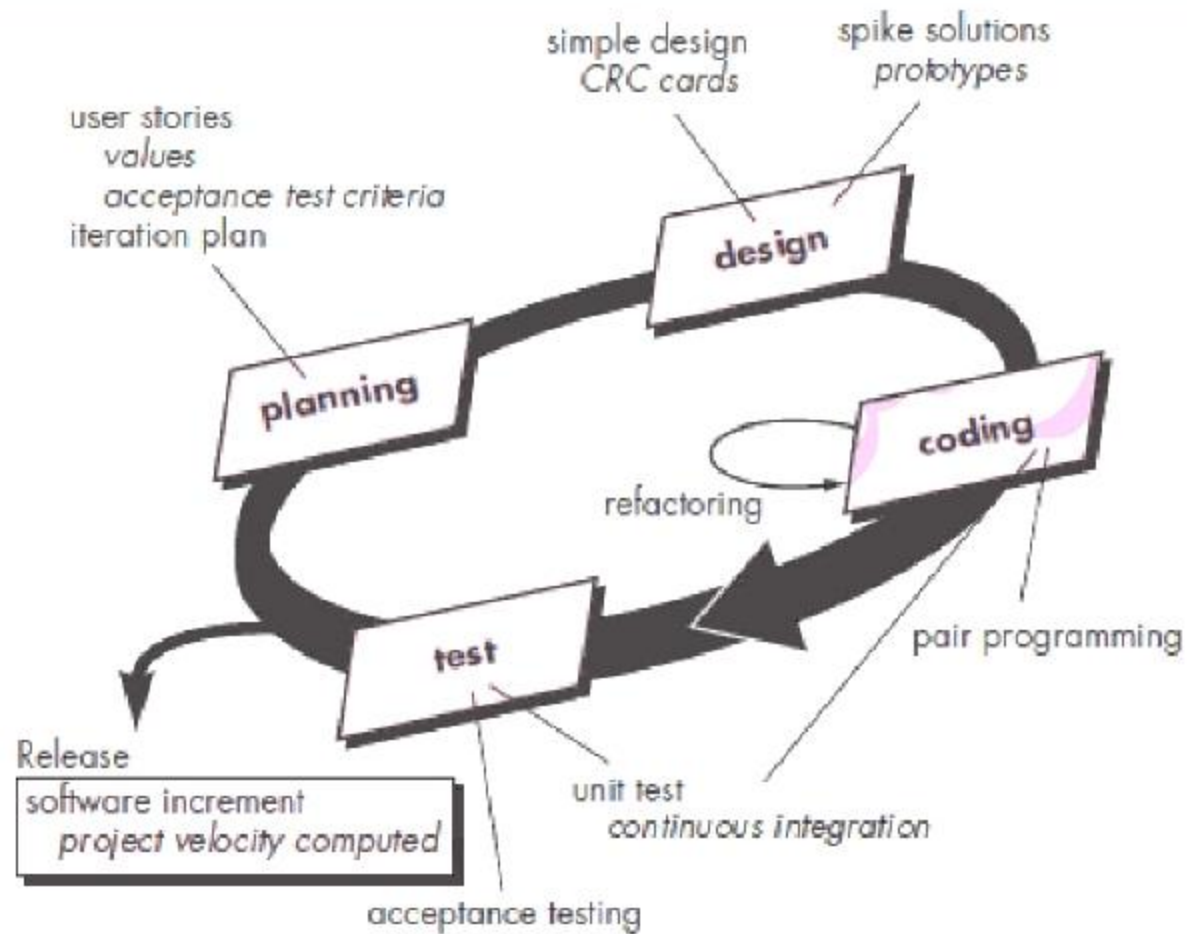
۳- اولویت بندی با توجه به مقدار ریسک

در این مرحله سرعت پروژه نیز با توجه به نیازهای مشتری محاسبه می شود که:

۱- کمک به برآورد تاریخ های تحویل و زمان بندی

۲- جلوگیری از زیاده روی در یک مورد

فرایند xp (طراحی)



طراحی XP از اصل KIS پیروی می کند.

کارت های CRC (کلاس-مسئولیت-همکار)

راهکار خیزشی (spike solution)

بازآرایی

محصول گذرا

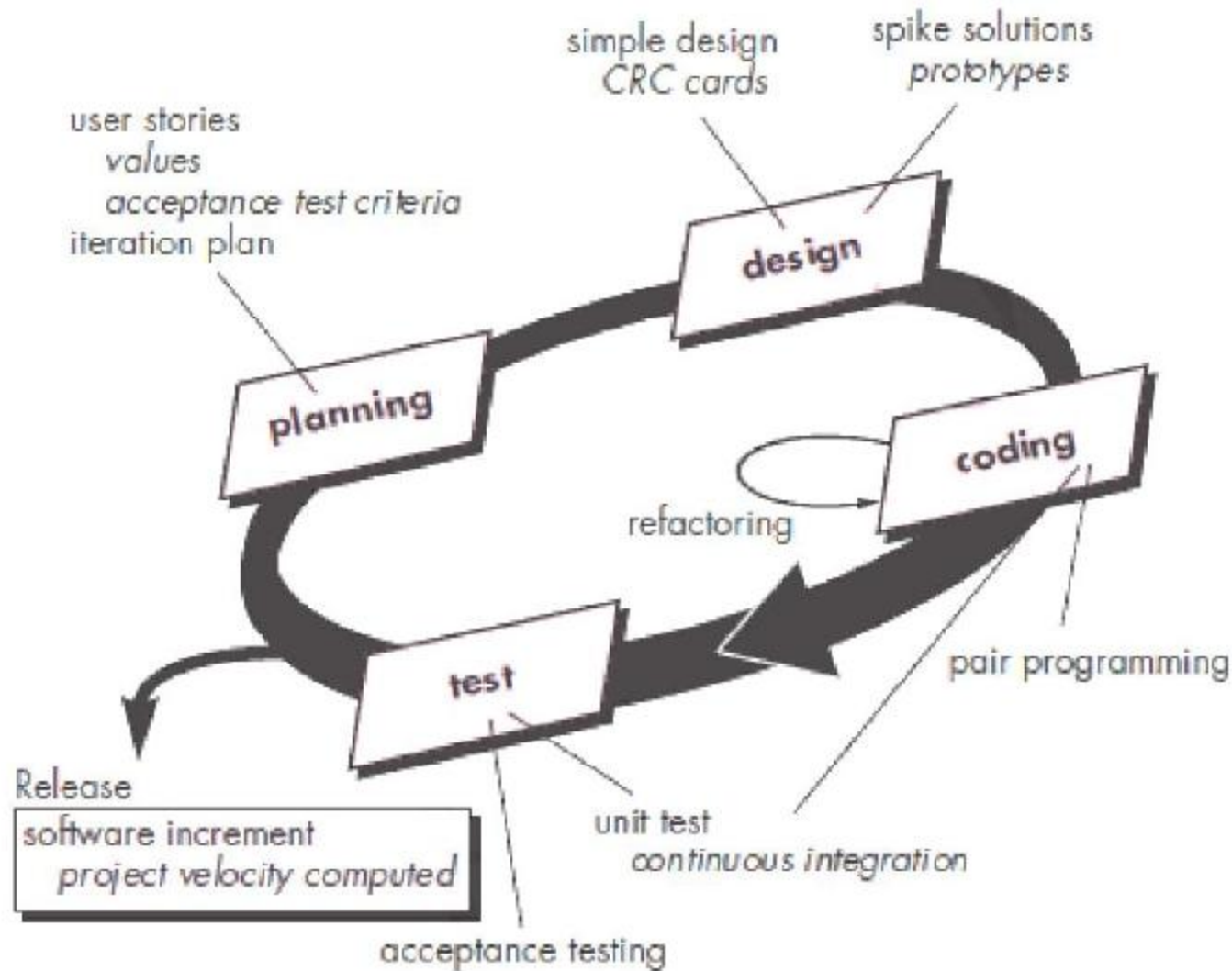
فرایند xp (کدنویسی)

پس از توسعه یافتن داستان ها و انجام شدن کارهای طراحی مقدماتی تیم به کد نویسی نمی پردازد بلکه یک سری آزمون واحد تهیه می کند.

برنامه نویسی جفتی :

دو نفر با هم روی یک ایستگاه کاری کار کنند و کد مربوط به یک داستان را بنویسند.

فرایند xp (آزمون)



- آزمون واحدها قبل از شروع کدنویسی

- آزمون های پذیرش Xp (آزمون مشتری)

تکاملی ارگانیک از Xp است.

الهام گرفته از کمینه‌گرایی، مشتری‌مداری و آزمون‌مداری Xp.

بیشترین تفاوت IXP با XP

۱- اعمال مدیریت بیشتر

۲- گسترش نقش مشتریان

۳- ارتقای روش‌های فنی

نش عمل جدید IXP

۱- ارزیابی آمادگی :

پیش از شروع یک پروژه ی IXP ، سازمان باید ارزیابی آمادگی را انجام دهد. در این عمل اطمینان حاصل می شود که:

- ۱- یک محیط توسعه مناسب وجود دارد که IXP را پشتیبانی میکند
- ۲- تیم شامل مجموعه مناسبی از طرف های ذی نفع است.
- ۳- سازمان دارای یک برنامه کیفیت ممتاز است و بهبود مستمر را پشتیبانی می کند.
- ۴- فرهنگ سازمانی، پشتیبان ارزش های جدید یک تیم جدید است.
- ۵- جامعه ی وسیعتر پروژه از افراد مناسبی تشکیل شده است.

نقش عمل جدید IXP

۲- جامعه پروژه:

وقتی که قرار باشد که برای یک پروژه مهم در سازمانی بزرگ از روش Xp استفاده شود مفهوم تیم باید به جامعه تغییر شکل پیدا کند.

این جامعه شامل :

یک متخصص فناوری و مشتریانی که در موفقیت پروژه نقش محوری دارند
و نیز شامل طرف های ذی نفع نظیر کارمندان حقوقی ، کارمندان بخش تولید و فروش و... باشد.

در IXP اعضای جامعه و نقش هرکدام از آنها باید به صراحت تعریف می شود.

۳- چارتر کردن پروژه :

تیم Ixp ، خود به ارزیابی پروژه می پردازد تا تعیین کند آیا یک توجیه تجاری مناسب برای پروژه وجود دارد و آیا پروژه اهداف و مقاصد کلی سازمان را پیش می برد یا خیر.

با عمل چارتر کردن ،حیطه پروژه برای تعیین چگونگی کامل شدن ،توسعه یافتن یا جایگزین ساختن سیستمها یا فرآیندهای موجود تعیین می شود.

۴- مدیریت مبتنی بر آزمون :

یک سری، مقاصد قابل سنجش و سازوکارهایی برای رسیدن به این مقاصد.

۵- بازنگری :

مرور و بازبینی مسائل رویدادها و درس‌های آموخته‌شده با هدف بهبود بخشیدن به فرآیند Ixp

۶- آموزش پیوسته :

چون آموزش بخش حیاتی از بهبود مستمر است، اعضای تیم XP تشویق می‌شوند تا روش‌ها و تکنیک‌های جدیدی را بیاموزند که به محصول با کیفیت بالاتر منجر شود.

مشاخره XP

۱- متغیر بودن خواسته ها

۲- نیازهای متناقض مشتریان

۳- خواسته ها به صورت غیر رسمی بیان می شود.

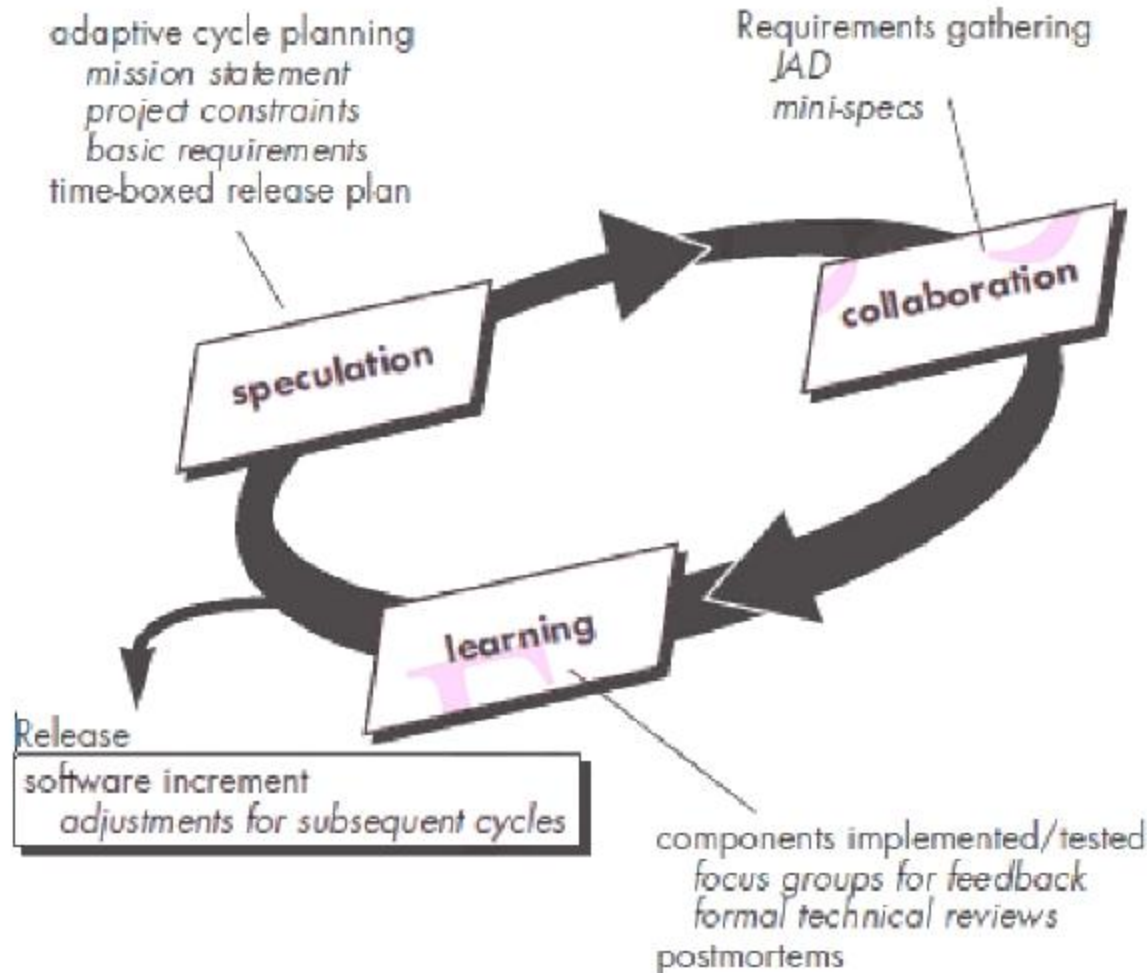
۴- فقدان طراحی رسمی

مدل های فرایند چابک

- ✓ برنامه نویسی حدی (XP)
- ✓ توسعه وفقی نرم افزار (ASD)
- ✓ روش توسعه سیستم های پویا (DSDM)
- ✓ کریستال
- ✓ توسعه ویژگی محور (FDD)
- ✓ توسعه نرم افزار ناب (LSD)
- ✓ مدل سازی چابک (AM)
- ✓ فرآیند یکپارچه چابک (AUP)

توسعه وفقی نرم افزار (ASD)

تکنیکی برای ساخت نرم افزارها و سیستم های پیچیده.
تاکید بر همکاری انسانی و خود سازمان دهی تیمی .



مراحل چرخه حیات ASD

- تعمق (تفکر)
- همکاری
- یادگیری



اسکرام (Scrum)

اصول اسکرام با بیانیه چابکی سازگاری دارد.

شامل فعالیت‌های چارچوبی زیر است :

خواسته‌ها ، تحلیل ، طراحی ، تکامل ، تحویل.

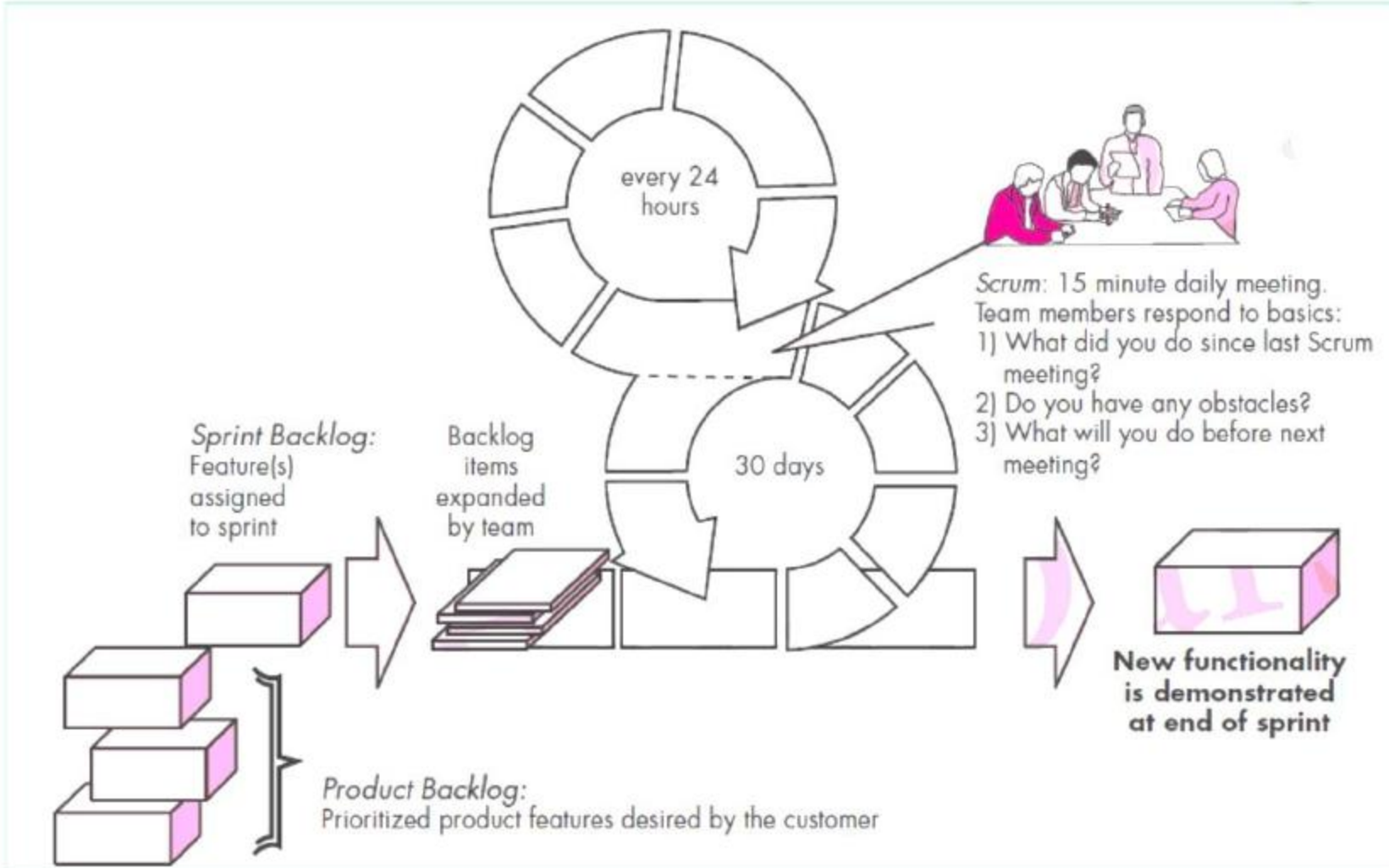
تاکید اسکرام ...

• فهرست جامانده‌ها (Backlogs)

• sprint ها .

• نشست‌های اسکرام.

• دموها.



روش توسعه سیستم‌های پویا (DSDM)

Dynamic Systems Development Method (DSDM)

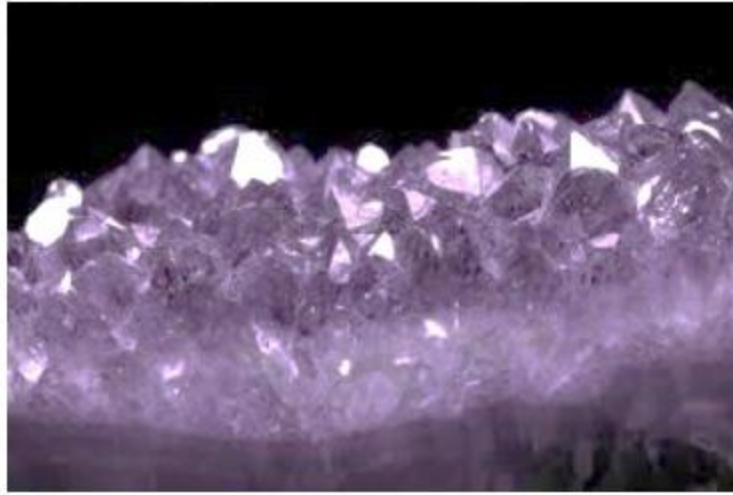
چارچوبی بر ساخت و نگهداری سیستم‌هایی فراهم می‌آورد که قید و بندهای زمانی فشرده را از طریق به کار گیری نمونه‌ی اولیه در یک محیط پروژه کنترل شده برآورده می‌سازند.

چرخه حیات DSDM :

- ۱- **امکان‌سنجی** : آیا نرم‌افزار کاندیدای لایقی برای فرآیند DSDM هست یا خیر.
- ۲- **مطالعه تجاری** : خواسته‌های عملیاتی و اطلاعاتی را تعیین می‌کند که به برنامه کاربردی ارزش تجاری می‌دهند.
- ۳- **تکرار مدل‌های عملیاتی** : مجموعه‌ای از نمونه‌های اولیه افزایشی که عملکرد را برای مشتری به نمایش می‌گذارد. هدف این چرخه تکرار، جمع‌آوری خواسته‌های اضافی با توجه به بازخورد از کاربران در حین تمرین روی نمونه اولیه است.
- ۴- **تکرار طراحی و ساخت** : بازبینی نمونه‌های اولیه ساخته‌شده طی مرحله‌ی تکرار مدل‌های عملیاتی برای حصول اطمینان از اینکه هر کدام به شیوه‌ای مهندسی شده‌است که بتواند برای کاربران نهایی ارزش تجاری داشته باشد.
- ۵- **پیاده‌سازی** : قرار دادن آخرین نسخه نرم‌افزاری در محیط کار.

کریستال Crystal

کریستال به خانواده ای از مدل های فرایند چابک گفته می شود که بر خصوصیات ویژه ی یک پروژه قابل انطباق است.



توسعه ویژگی - محور

Feature Driven Development (FDD)

قدری رسمی تر از سایر روش های چابک است، ولی همچنان با جلب توجه تیم پروژه به توسعه ی یک سری ویژگی ها چابکی را حفظ می کند.

فلسفه :

- (1) emphasizes col- laboration among people on an FDD team;
- (2) manages problem and project complexity using feature-based decomposition followed by the integration of software increments
- (3) communication of technical detail using verbal, graphical, and text-based means.

ویژگی *feature*

ویژگی : یک عملکرد است که نزد متقاضی دارای ارزش بوده در کمتر از دو هفته قابل پیاده سازی است.

قالب تعریف ویژگی

<action> the <result> <by | for | of | to> a(n) <object>

- *Add the product to shopping cart*
- *Display the technical-specifications of the product*
- *Store the shipping-information for the customer*

<action><-ing> a(n) <object>

Making a product sale

توسعه نرم افزار ناب

Lean Software Development (LSD)

اصول تولید ناب را وارد دنیای مهندسی نرم افزار کرده است.

- *eliminate waste*
- *build quality in*
- *create knowledge*
- *defer commitment*
- *deliver fast*
- *respect people*
- *optimize the whole.*

تفسیر حذف ضایعات در حیطه یک پروژه نرم افزاری چابک

- 1- adding no extraneous features or functions
- 2- assessing the cost and schedule impact of any newly requested requirement
- 3- removing any superfluous process steps
- 4- establishing mechanisms to improve the way team members find information,
- 5- ensuring the testing finds as many errors as possible
- 6- reducing the time required to request and get a decision that affects the software or the process
that is applied to create it
- 7- streamlining the manner in which information is transmitted to all stakeholders involved in the process.

مدل سازی چابک

Agile Modeling (AM)

پیشنهاد می کند که مدل سازی برای همه سیستم ها ضروری است، ولی پیچیدگی ، نوع و اندازه مدل باید متناسب با

نرم افزاری که قرار است ساخته شود، تنظیم گردد.

- (1) all constituencies can better understand what needs to be accomplished
- (2) the problem can be partitioned effectively among the people who must solve it
- (3) quality can be assessed as the system is being engineered and built.

اصولی که AM را از سایر روش ها متمایز می سازد

Model with a purpose.

Use multiple models.

Travel light.

Content is more important than representation.

Know the models and the tools you use to create them.

Adapt locally.

فرایند چابک یکپارچه

Agile Unified Process (AUP)

فلسفه ترتیب در مقیاس انبوه و تکرار در مقیاس کوچک را بر ساخت نرم افزار مطرح می سازد.

1- Modeling.

2- Implementation.

3- Testing.

4- Deployment

5- Configuration and project management.

6- Environment management.

در نظر گرفتن فرایند چابک

SAFEHOME



Considering Agile Software Development

The scene: Doug Miller's office.

The Players: Doug Miller, software engineering manager; Jamie Lazar, software team member; Vinod Raman, software team member.



پایان فصل ۳