



Extreme Programming(XP)

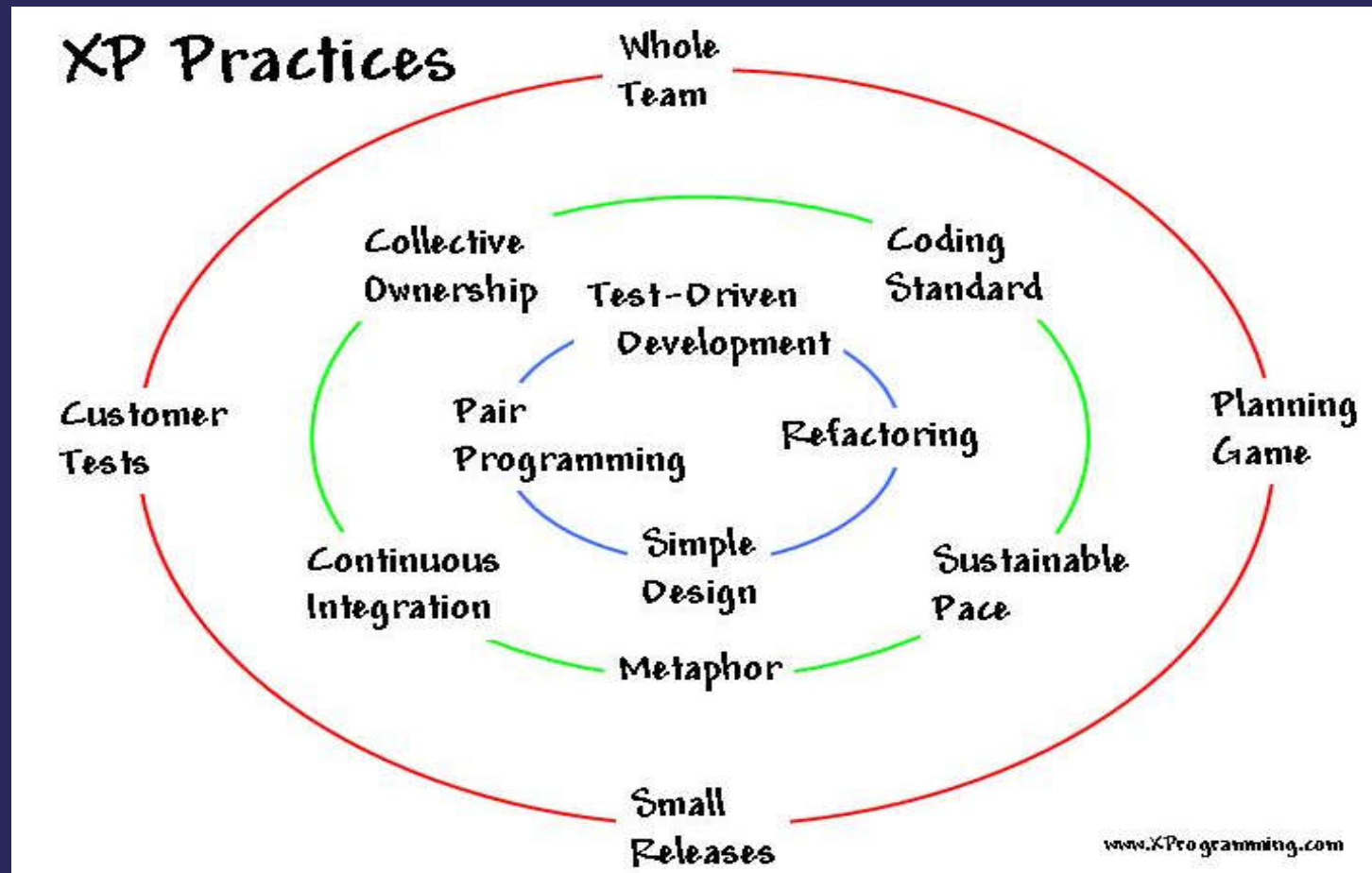


XP

- XP شناخته شده ترین و پر استفاده ترین روش agile است که در سال ۲۰۰۰ توسط Kent Beck پیشنهاد داده شده است.
- در XP نیازمندی‌های کاربران به صورت سناریوها یا user story ها بیان می‌شود.
- User story ها در قالب مجموعه ای از task ها پیاده‌سازی می‌شوند.
- برنامه‌نویسان به صورت گروه‌های دو نفره (pairs) task ها را پیاده‌سازی می‌کنند. پیش از پیاده‌سازی هر task برای آن تست‌های واحد تولید می‌شوند.
- پیش از اینکه که بخش های پیاده‌سازی شده با سیستم یکپارچه شوند تمام تست‌ها باید انجام شده باشد.



XP Practices





XP Practices : whole team

➤ تمام مشارکت کنندگان در پروژه‌ی XP، اعضای تیم هستند. در XP تیم تولید، علاوه بر تولیدکنندگان شامل یک نماینده از کسب‌وکار (مشتري) می‌باشد که نیازمندی‌ها را فراهم کرده، اولویت‌ها را تعیین و پروژه را راهنمایی می‌کند. علاوه بر این مشتري در تعریف تست های پذیرش نیز شرکت دارد.



XP Practices : Planning the game

Release planning ➤

- نیازمندی‌های مشتریان به همراه تیم تولید از طریق سناریوها مورد بررسی قرار می‌گیرد. سپس این نیازمندی‌های به صورت user story ها مشخص شده و در story card ها ثبت می‌گردد.
- مشتریان بر مبنای ارزش هر story به آن‌ها اولویت می‌دهند. (value)
- اعضای تیم هزینه مورد نیاز برای پیاده‌سازی هر story را (بر اساس میزان هفته) برآورد می‌کنند، در صورتی که هزینه بیش از میزان مورد نظر باشد، از مشتری خواسته می‌شود تا story ها را به story های کوچکتر بشکند و تخصیص اولویت و هزینه دوباره انجام می‌پذیرد. (split story)



XP Practices : Planning the game

Iteration planning ➤

- وظایف و فعالیت‌های مربوط به تولیدکنندگان سیستم (developers) برای هر دو هفته در آن برنامه ریزی می‌شود. (هر iteration حدود دو هفته طول می‌کشد).
- در طی Iteration planning مشتریان و تیم تولید مشخص می‌کنند چه story هایی در iteration بعد قرار گیرند.
- تیم تولید story ها را به task های مختلف تقسیم می‌کند و task ها نیز روی task card ها نوشته می‌شوند.
- task ها به برنامه نویسان اختصاص داده می‌شوند و زمان تکمیل آن‌ها نیز تخمین زده می‌شود.



Sample story card template



As a (role) I want (something) so that (benefit).

Id : us_10

Title: Always keep contact list updated

Description: As a user, I want to add my contacts in the instant messaging app (like telegram) so that this contacts' data also added to contacts in my phonebook.

Conformation: If I change the name of a contact in the app, its related should be updated in phonebook.

Related Stories:

Author: "M.hashemi"

Date Created:

Priority: High

Status: "in progress"

Estimated effort: 3 person /hour

Rough size : Medium



Example of story card for “document downloading”



Case: Downloading and printing an article

First, you select the article that you want from a displayed list. You then have to tell the system how you will pay for it - this can either be through a subscription, through a company account or by credit card.

After this, you get a copyright form from the system to fill in and, when you have submitted this, the article you want is downloaded onto your computer

You then choose a printer and a copy of the article is printed. You tell the system if printing has been successful.

If the article is a print-only article, you can't keep the PDF version so it is automatically deleted from your computer.



Example of Task card for “document downloading”



Task 1: Implement principal workflow

Task 2: Implement article catalog and selection

Task 3: Implement payment collection

Payment may be made in 3 different ways. The user selects which way they wish to pay . If the user has a library subscription, then they can input the subscriber key which should be checked by the system. Alternatively, they can input an organizational account number . If this is valid, a debit of the cost of the article is posted to this account. Finally, they may input a 16 digit credit card number and expiry date. This should be checked for validity and, if valid a debit is posted to that credit card account.



XP Practices : Small release

- ابتدا در اولین ترخیص، مجموعه‌ای حداقل از کارکردهای مفید (useful functionalities) که ارزش کسب‌وکاری (business value) فراهم می‌کند تولید می‌شود. ترخیص‌های سیستم به طور مکرر و افزایشی کارکردهای جدید را به ترخیص اولیه اضافه می‌کند.
- به عبارتی در هر iteration، نرم افزار تست شده که ارزش های کسب‌وکاری مورد نظر را فراهم می‌آورد به مشتری داده می‌شود. مشتری می‌تواند از این نرم افزار برای هر منظوری استفاده کند (می‌تواند فقط آن را ارزیابی کند یا آن را به کاربر نهایی تحویل دهد). مهم‌ترین نکته این است که نرم افزار visible بوده و در پایان هر iteration به مشتری تحویل داده می‌شود.



XP Practices : Acceptance customer test



- تست‌های پذیرش که تست مشتری نیز خوانده می‌شوند توسط مشتری طراحی می‌گردند و بر ویژگی‌های کلی سیستم و کارکردهایی که توسط مشتری قابل مشاهده و بررسی هستند تمرکز دارند.
- تست‌های پذیرش از user story هایی که به عنوان بخشی از یک release پیاده سازی شده‌اند به دست می‌آیند.
- تست‌های پذیرش تست‌های جعبه سیاه هستند و هر کدام دارای نتایج مورد انتظار از سیستم می‌باشند.



XP Practices : Simple design

- تیم XP، طراحی را ساده و به اندازه‌ی کافی انجام می‌دهد نه بیشتر.
(simple and adequate)
- بر مبنای قانون keep it simple
- اگر لازم باشد تا طراحی بهبود پیدا کند، بعداً روی آن refactoring انجام می‌شود.
- Refactoring به مهندس نرم‌افزار این امکان را می‌دهد که بدون تغییر کارکرد یا رفتار بیرونی (external functionality or behavior) ساختار داخلی طراحی یا کد را بهبود بخشد. به این ترتیب refactoring برای بهبود کارایی، خوانایی و عملکرد طراحی یا کد مورد استفاده قرار می‌گیرد.



XP Practices : Refactoring

- Design refactoring که در اسلاید پیش به آن اشاره شد.
- Code refactoring
 - تمام برنامه نویسان باید به طور مداوم روی کد، به محض اینکه امکان بهبود آن وجود داشت refactoring را انجام دهند. (حتی اگر نیازی نباشد).
- مثالهایی از refactoring عبارتند از: سازمان دهی مجدد سلسله مراتب کلاسها برای حذف duplicate code، تغییر نام صفات و متدها، جای گذاری فراخوانی متدها به جای خطوط کد برنامه و ...



XP Practices : Pair Programming

- Pair programming یکی از practice های ابتکارانه در XP می باشد که در آن برنامه نویسان به صورت دو نفره برای تولید نرم افزار کار می کنند. در این روش دو برنامه نویس وظیفه تولید را با همکاری هم بر روی یک Workstation بر عهده دارند.
- در Pair programming همواره دو برنامه نویس با هم کار نمی کنند، بلکه زوج های دو نفره به صورت پویا ساخته می شوند. بنابراین تمام اعضای تیم در تولید فرآیند تولید با یکدیگر کار می کنند.



XP Practices : Pair Programming

➤ فواید Pair programming به قرار زیر است:

- به مالکیت مشترک (common ownership) کد کمک می‌کند.
- به عنوان یک فرآیند informal review عمل می‌کند که در آن هر خط از کد توسط حداقل دو نفر مورد بررسی قرار می‌گیرد.
- موجب پشتیبانی از refactoring کد می‌گردد.
- بهره‌وری تولید با به کار گیری pair programming قابل مقایسه با حالتی است که دو نفر به طور مستقل کار کنند.



XP Practices : Test-first development



- نوشتن تست پیش از آغاز کد نویسی یکی از ابتکارات XP است. به جای اینکه ابتدا کد نوشته شود و سپس تست‌هایی برای آن تولید شوند، تولید تست پیش از شروع کد نویسی انجام می‌گیرد.
- نوشتن تست به این صورت به طور ضمنی موجب تعریف و واضح تر شدن رفتار کارکردهای مورد نظر برای پیاده سازی و همچنین interface می‌شود. به این ترتیب مشکلات نیازمندی‌ها (ابهامات ، از قلم افتادگی‌ها و ...) و واسط‌ها پیش از پیاده سازی شناسایی می‌شود.
- مانع از بروز test lag (تاخیر در تست) می‌شود. زیرا test lag زمانی اتفاق می‌افتد که تولید کننده سیستم با سرعتی بیشتر از تست کننده ی سیستم کار خود را پیش می‌برد. از این رو ممکن است با پیش افتادن تولید از تست، برخی از تست‌ها از قلم بیافتند.
- برای نوشتن تست‌های واحد از یک چارچوب تست واحد خودکار استفاده می‌شود.



XP Practices : Coding standard, Collective code ownership



Coding standard ➤

- تیم XP یک استاندارد کد زنی مشترک را دنبال می کنند بنابراین طوری به نظر می رسد که تمام کد سیستم را یک نفر نوشته است.
- این موجب پشتیبانی از collective ownership می گردد.

Collective code ownership ➤

- تمام افراد تیم در تمام بخش های کد سیستم کار می کنند بنابراین تمام آن ها مالک کد هستند نه یک فرد. از این رو هر فرد می تواند هر بخش از کد را که لازم باشد تغییر دهد.



XP Practices : Metaphor, Sustainable pace



Metaphor ➤

- تیم XP از یک روش نامگذاری مشترک برای کلاس ها، متدها و ... استفاده می کنند تا مطمئن شوند تمام افراد نحوه ی کار سیستم را متوجه می شوند. علاوه براین افراد می دانند که یک کارکرد مورد نظر را کجا پیدا کنند و یا آن را به کجا اضافه کنند.

Sustainable pace ➤

- اعضای تیم XP، نباید بیش از ۴۰ ساعت در هفته کار کنند، در صورت نیاز به اضافه کاری در یک هفته ، هفته ی بعد نباید شامل اضافه کاری باشد.
- به عبارتی اعضای تیم تنها در صورتی که موثر باشد اضافه کاری می کنند.
- این مفهوم ناشی از این مطلب است که افراد وقتی دارای زمان استراحت کافی باشند بهتر کار می کنند و در نتیجه کیفیت کار بالاتر می رود.



XP Practices : Continuous integration



- تیم XP سیستم را همیشه یکپارچه نگه می‌دارد، به محض اینکه کار یک task به پایان می‌رسد، آن را با کل سیستم یکپارچه می‌کند.
- یکپارچه سازی مداوم مانع از ایجاد مشکل پخش و تکه تکه شدن و عدم امکان استفاده مجدد از کدهای تولید شده می‌گردد.
- هر فرد باید بر روی آخرین نسخه کار کند.
- یکپارچه سازی مداوم مانع از بروز تاخیراتی در پروژه می‌شود که ناشی از مشکلات یکپارچه سازی هستند.
- یکپارچه سازی مداوم موجب تشخیص و مانع از بروز مشکلات سازگاری بین کدها می‌شود.
- بعد از هر گونه یکپارچه سازی باید تمام تست های واحد سیستم pass شوند.



XP practice and Agile principles

principle	توضیح
Customer involvement	از طریق همراهی مداوم مشتری با تیم تولید انجام می پذیرد. یک نماینده مشتری در تولید شرکت داده می شود که از طریق آن نیازمندی ها شناسایی و تست های پذیرش تعریف می گردند.
Incremental delivery	از طریق ترخیص های کوچک و مکرر پشتیبانی می شود. کارکردهایی که در هر increment سیستم قرار می گیرند بر مبنای user story ها و سناریوهایی می باشد که نیازمندی های کاربر را در بر می گیرند.
People not process	از طریق pair programming، collective code ownership، sustainable pace پشتیبانی می گردد.
Embrace change	از طریق ترخیص های منظم سیستم به مشتری، test first development، refactoring و یکپارچه سازی مداوم کارکردهای جدید پشتیبانی می شود.
Maintain simplicity	از طریق refactoring کد که موجب بهبود کیفیت آن می شود و هم چنین با طراحی ساده که تغییرات آینده غیر ضروری را پیش بینی نمی کند پشتیبانی می شود.

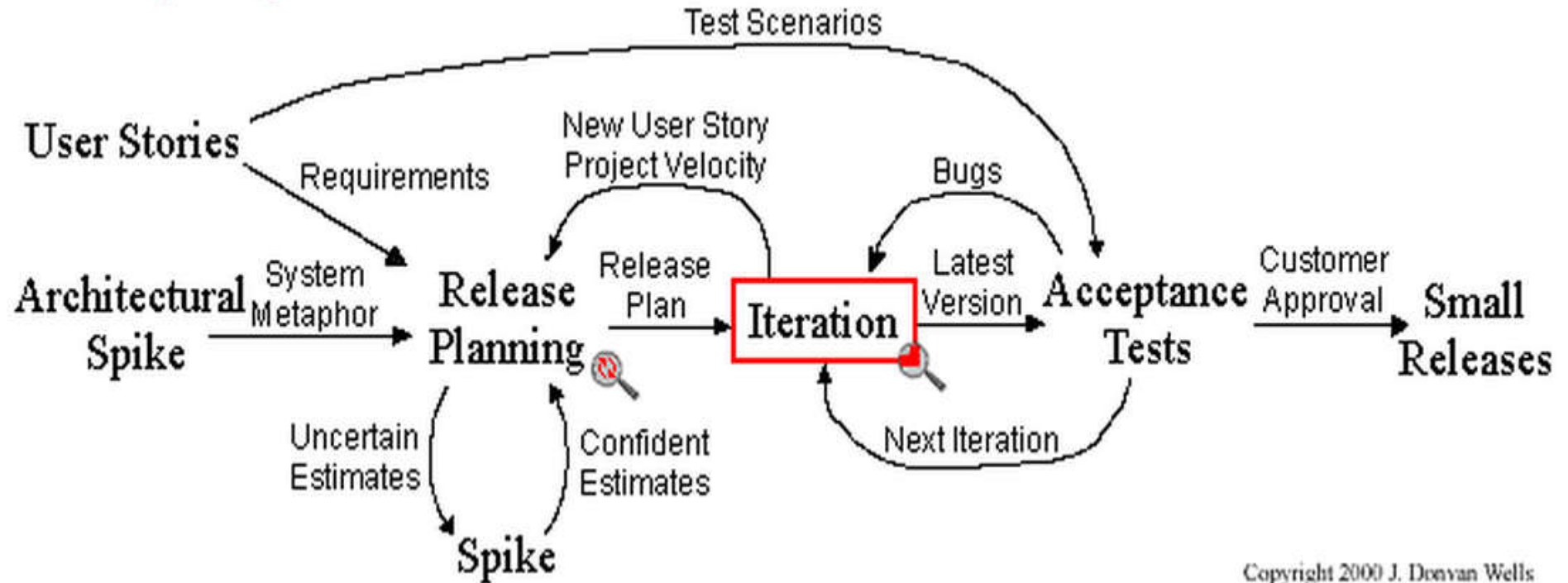


XP and change

- یک اصل متعارف در مهندسی نرم افزار طراحی برای تغییر است. با توجه به اینکه پیش‌بینی تغییرات هزینه‌های بعدی در چرخه‌ی حیات را کاهش می‌دهد، صرف زمان و تلاش برای آن ارزشمند است.
- با این وجود XP این اصل را کنار می‌گذارد و اعتقاد دارد که طراحی برای تغییر هدر دادن تلاش است. زیرا نمی‌توان تغییرات را به طریقی قابل اطمینان پیش‌بینی نمود. (ممکن است تغییراتی کاملاً متفاوت با آنچه پیش‌بینی شده در خواست شوند).
- به این ترتیب XP، تغییرات را می‌پذیرد و با انجام refactoring مداوم در صورت لزوم اعمال تغییرات پس از رخداد آن‌ها، این امر را تسهیل می‌بخشد.



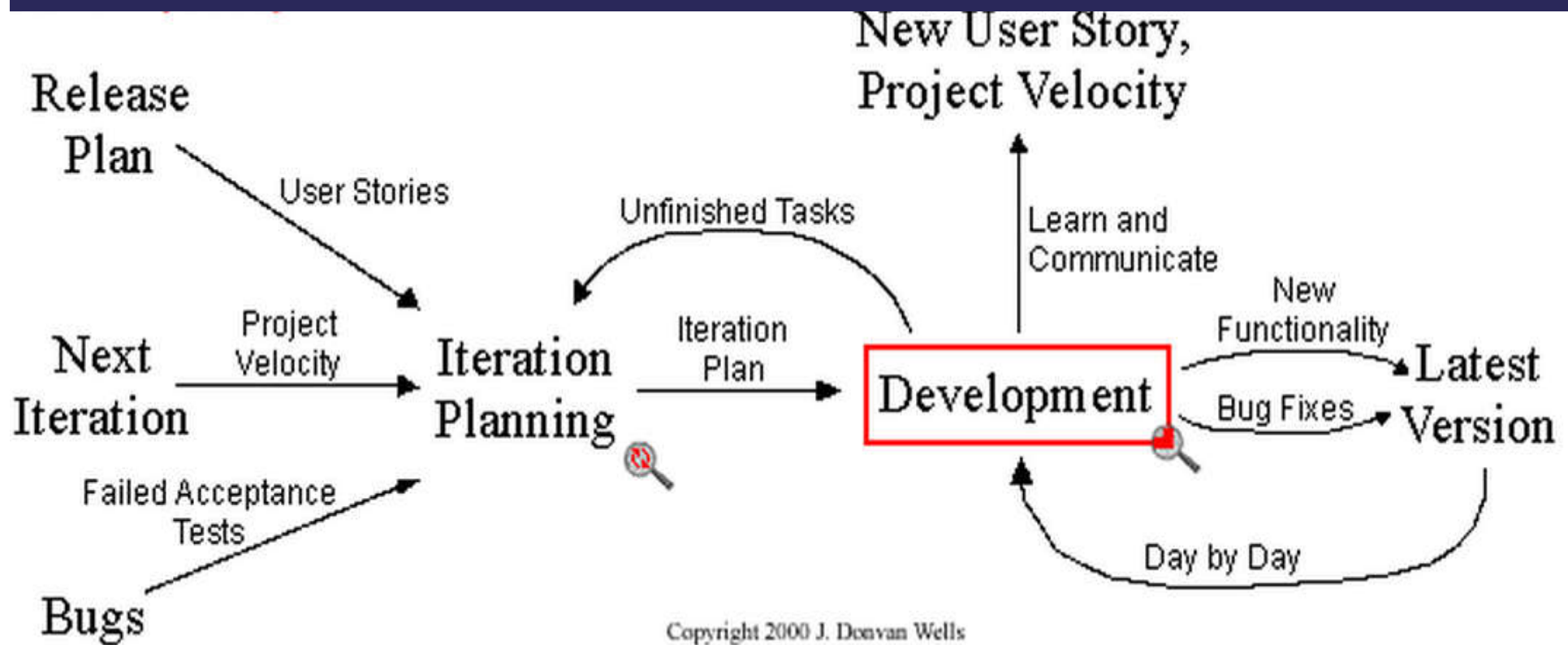
XP Practices



Copyright 2000 J. Donovan Wells

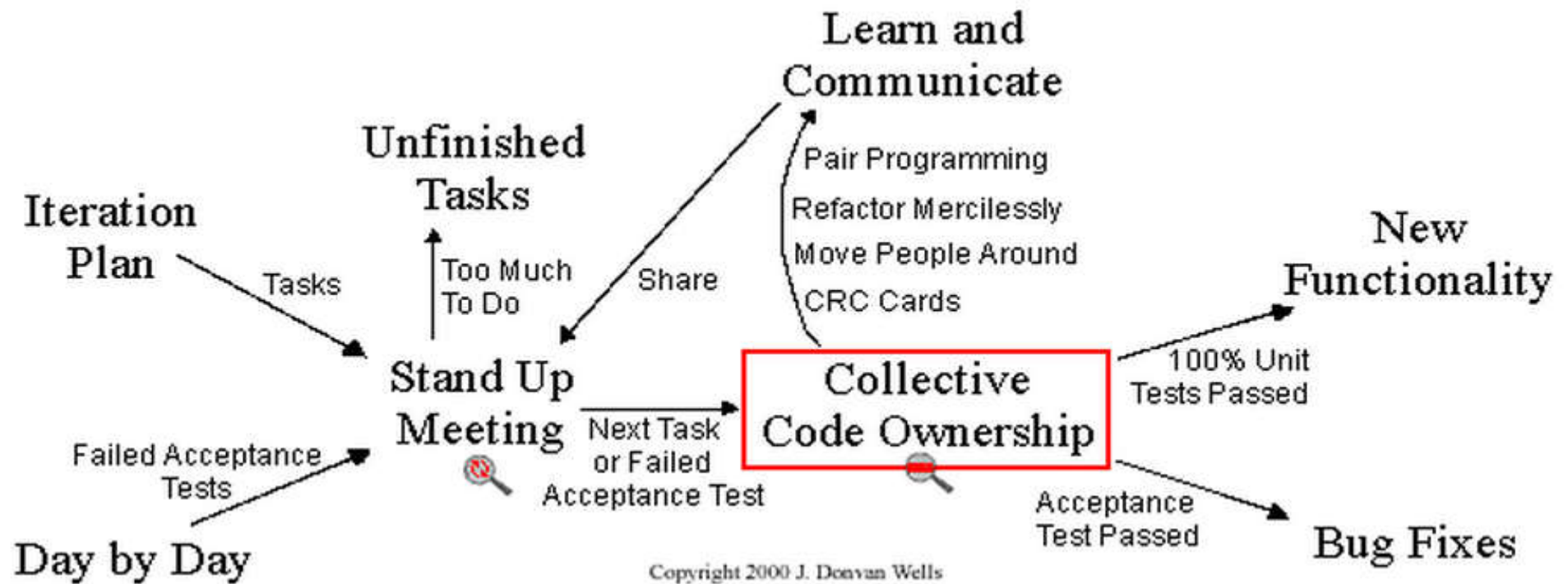


XP Practices



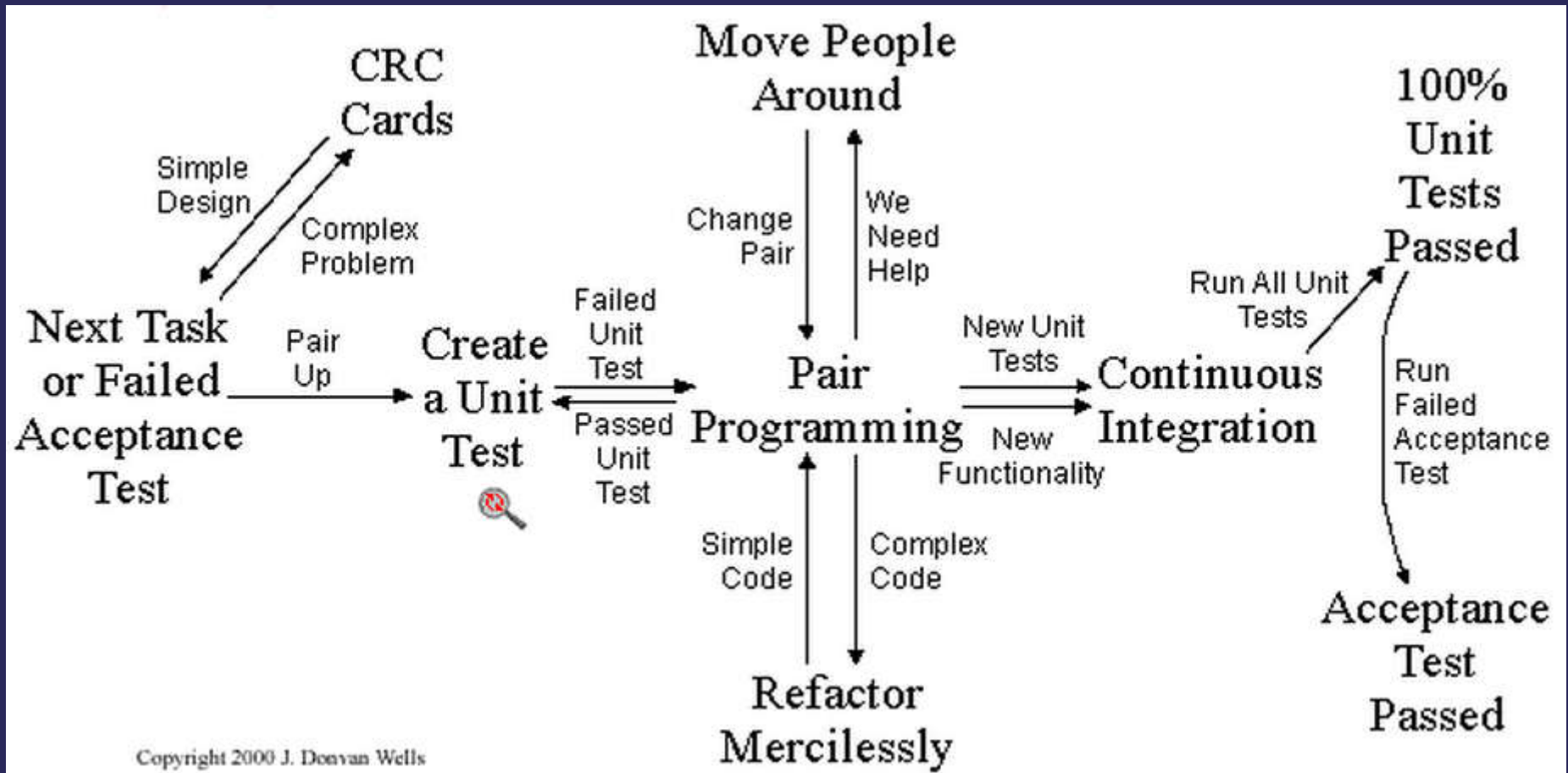


XP Practices



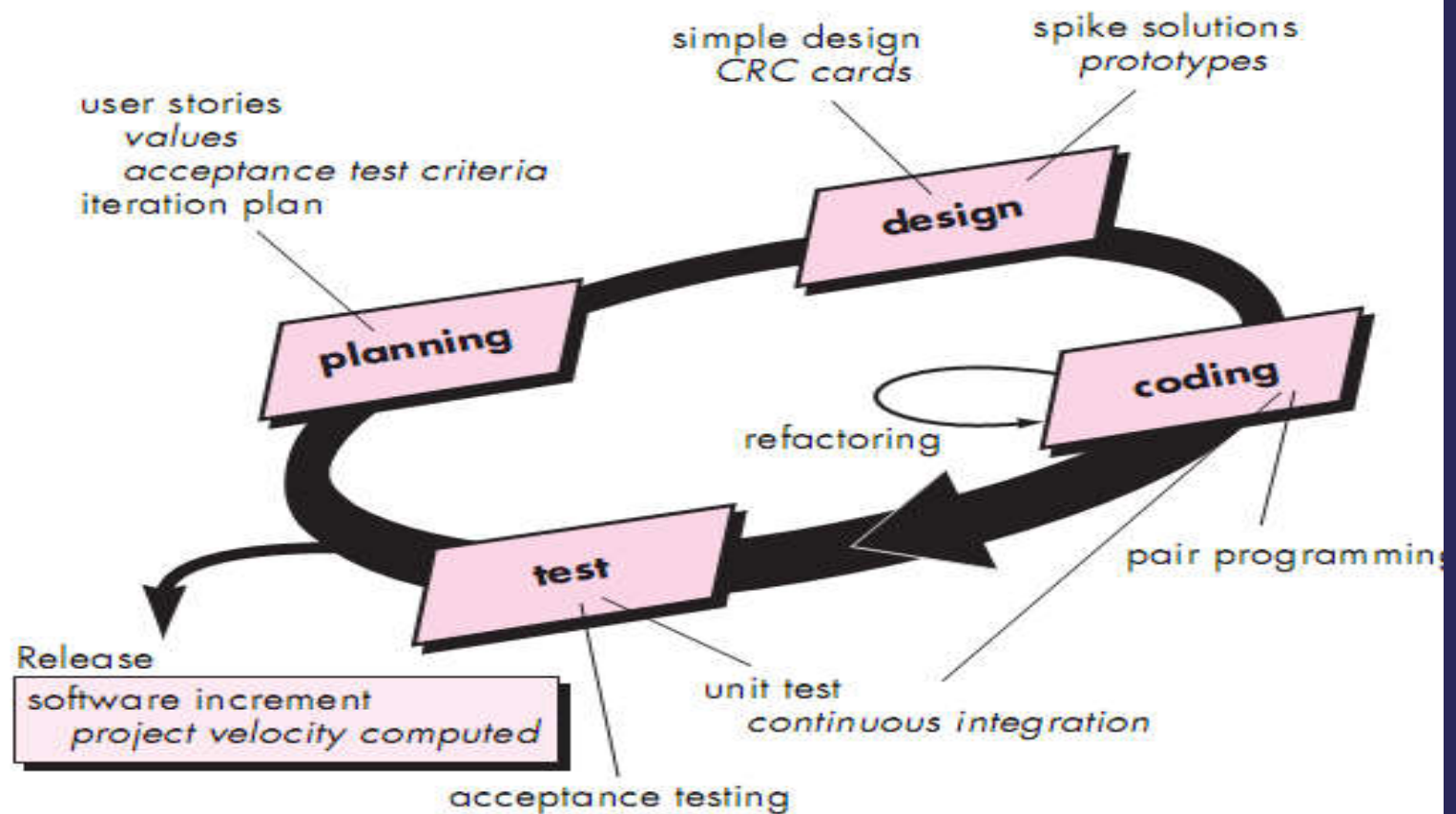


XP Practices





فرآیند XP





طراحی در XP

- بر مبنای قانون KIS (Keep it Simple)
 - طراحی برای کارکردهای اضافه (extra functionalities) انجام نمی‌پذیرد.
- توصیه به استفاده از کارت‌های (collaborator Class Responsibility) CRC
 - با توجه به اینکه XP، از رویکرد شی گرای استفاده می‌کند، استفاده از CRC کارت‌ها توصیه می‌شود. به وسیله ی این کارت کلاس های مربوط به increment فعلی شناسایی و سازمان دهی می‌شوند.
- کارت های CRC تنها محصول کاری طراحی هستند که در فرآیند XP تولید می‌شوند.
- اما برای مسائل طراحی پیچیده تولید prototype توصیه می‌شود. این prototype های طراحی spike solution خوانده می‌شوند.



طراحی در XP (ادامه)

➤ توصیه به انجام Refactoring برای بهبود طراحی



کد نویسی در XP

- توصیه بر تولید تست های واحد (unit test) پیش از شروع کد زنی
- توصیه بر pair programming



تست در XP

- تولید unit test از story ها قبل از شروع به کدزنی
- انجام unit test خودکار به صورت روزانه
- acceptance test توسط کاربر طراحی و توسط تیم تولید انجام می شود.

■



Test case description

Test 4: Test credit card validity

Input:

A string representing the credit card number and two integers representing the month and year when the card expires

Tests:

Check that all bytes in the string are digits

Check that the month lies between 1 and 12 and the year is greater than or equal to the current year

Using the first 4 digits of the credit card number check that the card issuer is valid by looking up the card issuer table. Check credit card validity by submitting the card number and expiry date information to the card issuer

Output:

OK or error message indicating that the card is invalid



مشکلات استفاده از XP

Customer involvement ➤

- پیدا کردن یک مشتری که نماینده ای از سایر ذی نفعان بوده و بتواند عضوی از تیم XP شود دشوار می باشد.

Architecture design ➤

- شیوهی افزایشی تولید موجب اتخاذ تصمیمات نامناسب برای معماری در مراحل ابتدایی فرآیند می شود.
- مشکلات و مسائل ناشی از این گونه تصمیمات ممکن است زمانی مشخص شود که feature های بسیاری پیاده سازی شده باشند و هزینهی refactoring معماری بسیار بالا باشد.



مشکلات استفاده از XP

Test complacency ➤

- با توجه به تعداد زیاد تست‌ها، تیم به راحتی متقاعد می‌شود که سیستم به طور مناسب تست شده است.
- به دلیل استفاده از تست خودکار، تولید تست به سمت تست‌هایی که ساده تر به صورت خودکار انجام می‌شود متمایل است تا تولید تست‌های خوب.



منابع

- R. S. Pressman, *Software engineering: a practitioner's approach*. McGraw-Hill Higher Education, 2010.
- S. Ian, *software engineering, 9 th education*. Addison–Wesley, Ed, 2011.
- <http://www.extremeprogramming.org/>