بررسی اجمالی روند تولید نرم افزار به روش XP

برگرفته از :

http://softprojects.org/fa/KB/Articles/Analysis/Extreme-Programming-process-review.aspx

XPیا Extreme Programming در واقع یک فرآیند توسعه نرم افزار عمیق و منظم می باشد. این روش از سال ۱۹۹۰ توسط شخصی به نام Kent Beck به همراه Ward Cunningham این فرآیند را برای توسعه آسان نرم افزارها ایجاد و در سالهای بعد آن را تکمیل کردند به نحوی که از سال ۱۹۹٦ به عنوان یک روش مناسب کاربردهای خود را نشان داد و هم اکنون در شرکتهای مختلفی با سایز های متفاوت مورد استفاده می باشد. یکی از دلایل موفقیت این روش تاکید آن بر رضایت مشتری است. این متدلوژی برای ارائه چیزی که واقعا مشتری نیاز دارد طراحی شده است. همچنین این روش کمک می کند که نیازهای مشتری را حتی در پایانی ترین مراحل تولید در سیستم اعمال کنند. از دیگر تاکید های روش توجه به کار گروهی است و این کار را با ساده ترین و مؤثرترین راه انجام می دهد.. مدیران، مشتریان و توسعه دهندگان همه اعضای تیمی هستند که مختص تحویل یک نرم افزار خوب (با کیفیت) ایجاد شده است.

XPیک پروژه نرم افزاری را در چهار وجه ، ارتباطات (Communication) ، سادگی (Simplicity)، بازخورد ها (Feedback) و شجاعت (Courage) بهبود می بخشد: ۱-برنامه نویس XP ابتدا با مشتری ارتباط بر قرار می کند، سپس برنامه سازی را دنبال می کند 2- . آنها طراحی خود را ساده و تمیز نگه می دارند. ۳- با آزمایش نرم افزارهای خود ار روز اول بازخورد می گیرند. ٤- سیستم را در اولین فرصت به مشتری تحویل می دهند و تغییرات را

بـه محـض پیشـنهاد دادن انجـام مـی دهنـد. بـر پایـهXP ، برنامـه نویسـان قـادر خواهنـد بـود کـه شـــــــجاعانه بـــــــه تغییـــــــرات نیازهـــــــا و فنـــــــآوری پاســـــــخ دهنـــــــد. XPشـامل اجـزا زیـاد کـوچکی اسـت کـه در نگـاه اول هـر کـدام معنـی خاصـی نمـی دهـد امـا وقتـی بایکـدیگر ترکیـب شـدند یـک تصـویر کامـل مـی سـازند. ایـن یـک فاصـله اساسـی بـا توسـعه سـنتی نرم افزارها ایجاد می کند. در یک کلام XP متفاوت است.

مطلـب بـالا خلاصـه اسـت از ادعاهـایی کـه طراحـان XP در مـورد روش خودشـان مطـرح کـرده انـد. در سلسله مطالبی، به مرور این روش خواهیم پرداخت.

اعضای تیم در پروژه

در روش اکس‌پـی مشـتری نرم‌افـزار بایـد جزئـی از تـیم اجرایـی پـروژه باشـد و برنامـه‌نویس و مشـتری بایـد بـا هـم کـار کننـد و از مشـکلات و نیازهـای هـم مطلـع باشـند. در اکس‌پـی مشـتری بـه کسـی یـا گروهـی گفتـه می‌شـود کـه نیازهـای برنامـه و اولویت‌هـای آن نیازهـا را خـوب می‌دانـد. در ایـن روش مشـتری و برنامه‌نویسـان بایـد در یـک اتـاق کـار کننـد(البتـه تبصـره‌ای نیـز در ایـن قسـمت وجـود دارد کـه می‌گویـد مشـتری می‌توانـد تـا حـداکثر پنجـاه متـر از برنامه‌نویسـان دور باشد.)

داستان‌های کاربران یاUser Stories

بـرای این‌کـه بتـوانیم بـرای پـروژه‌ای برنامـه‌ریزی کنـیم، بایـد از نیازهـای کـاربران مطلـع باشـیم. البتـه نیـازی نیسـت کـه همـه چیـز را از همـان اول بـدانیم و از جزئیـات نیازهـای کـاربران اطلاعـاتی کسـب کنـیم؛ زیـرا آن جزئیـات بـه احتمـال زیـاد در طـول پـروژه تغییـر پیـدا می‌کننـد. در اکس‌پـی بایـد در مـورد هـر نیـاز کـاربر یـا Requirement مقـداری بـا مشـتری صـحبت کنـیم و از مشـتری بخـواهیم چنـد کلمـه‌ای در مـورد هـر یـک از نیازهـا روی فـیش یـا کاغـذهای یادداشـت چسـب‌دار (Post-it) بنویسـد. همزمـان برنامـه‌نویس نیـز می‌توانـد برداشـت خـود را از آن موضـوع روی کاغـذی مشـابه

بنویسد و هر دو برگه را روی دیوار اتاق بچسبانند. این برگه‌ها می‌توانند باعث یادآوری اعضای تیم از نیازهای اولیه و همچنین سهولت در تخمین زمان و هزینه پروژه شوند.

دوره‌های زمانی کوتاه

پروژه اکس‌پی هر دوهفته یک بار یک قسمت از نرم‌افزار که کارایی بالایی دارد و قسمتی از نیازهای اولیه مشتری بوده است را تحویل می‌دهد و از مشتری در مورد آن قسمت نظرخواهی می‌شود و اگر نیاز بود، نظر مشتری سریعاً اعمال می‌گردد. به این دوره دو هفته‌ای Iteration نیز می‌گویند. هر Iteration قسمتی از نرم‌افزار را با توجه به User Storiesها تولید می‌کند که چند نیاز کاربر را برآورده می‌سازد. وقتی یک Iteration شروع شد، دیگر مشتری حق عوض کردن نیازهایی که بر سر آن‌ها توافق کرده است را ندارد.

تست های قبول شده

جزئیات نیازهای کاربران که در User Stories وجود دارد، در قالب (Acceptance Tests) AT که مشتری تعیین می‌کند برداشت می‌شود. این تست همزمان با Iteration می‌تواند انجام گردد و در قالب زبان‌های اسکریپتی نوشته می‌شود تا بتوان آن را چندین بار تکرار کرد. هدف اصلیAT ها تست کردن برنامه و حصول اطمینان از این است که آن قسمت از برنامه نوشته می‌شود که صددرصد نیاز مشتری را برآورده می‌سازد.

این تست‌ها هر وقت که قسمت جدیدی به برنامه اضافه می‌شود و برنامه کامپایل می‌گردد، دوباره اجرا می‌شوند و اگر اشکالی داشته باشند، پیغام خطا می‌دهند. در نتیجه وقتی سیستم به اتمام رسید، می‌توان مطمئن بود که سیستم بدون خطا است.

به علاوه، در اکس‌پی تمامی کدها به روش Test-Driven Development تولید می‌شوند . یعنی قبل از نوشتن هر قطعه کد، باید تست آن تولید شود و کاری کرد که کد در پاس کردن Unit Testناتوان باشد. بدین‌معنی که اول یک قطعه کد مثلاً برای افزودن دانشجوی جدید به

فهرست دانشجویان می‌نویسیم، ولی چون در آن کد قطعه‌ای از کد کلاسی که insert را انجام می‌دهد وجود ندارد، برنامه اشکال می‌گیرد. حال کار گروه اکس‌پی این است که برنامه را طوری درست کنند که این تست پاس شود و به قدری روی برنامه کار کنند تا برنامه دلخواه به دست آید.

برنامه نویسی دوتایی

معمولاً در اکس‌پی برنامه‌نویسان در گروه‌های دوتایی قرار می‌گیرند و وظیفه تکمیل قسمتی از برنامه را به عهده می‌گیرند. هر دو برنامه‌نویس در مورد هر کدام از نیازهای کاربران با هم بحث می‌کنند و قدم به قدم کلاس‌های برنامه را آماده می‌کنند .بدین ترتیب که در ابتدا کلاسی را به صورت خیلی ابتدایی و بدون هیچ طراحی اولیه به وجود میآورند و این کلاس را امتحان می کنند و در صورتی که این کلاس فاقد هر گونه اشکال باشد، کد اصلی برنامه را بر اساس این امتحان کلاس می‌نویسند. وقتی یکی از برنامه‌نویسان مشغول نوشتن قسمتی از برنامه است، برنامه‌نویس دیگر وظیفه کنترل صحت این کدها را عهده‌دار است و در صورت مشاهده هر گونه اشکال، نویسنده کد را مطلع می کند.

تیم همه کاره

در اکس‌پی همه اعضای تیم همه کار می‌کنند. همه رویGUI ، پایگاه اطلاعات و ... کار می کنند. این بدین منظور نیست که اعضای تیم نمی‌توانند در کار مشخصی حرفه‌ای باشند، بلکه همکاری در تمامی بخش‌های پروژه باعث خواهد شد که تجربه جدیدی کسب کنند . این تیم حق دارد کدهایی که دیگران نوشته‌اند را بارها چک کند، اشکالات آن را مشخص نماید و اگر اصلاحی دارد، آن را متذکر شود.

محیط کاری باز

تیم باید در مکانی باز کار کند. در اتاق پروژه باید میزهایی فراهم شوند که روی آن چند

کامپیوتر قرار دارد. در جلوی هر کامپیوتر هم باید دو صندلی برای اعضای Pair تعبیه شود. دیوارهای اتاق باید از نقشه‌ها و طرح‌های نرم‌افزاری به همراه UML مملو باشد. صدای اتاقی که در آن کار انجام می‌شود، معمولاً پر از زمزمه‌های اعضای تیم است. شاید بگویید که در محیطی پر زمزمه که نمی‌شود کار کرد. در جواب این سؤال باید گفت طبق تحقیقاتی که در دانشگاه میشیگان انجام شده است، راندمان کار در محیط‌های کاری‌ای که در آن زمزمه باشد و سکوت محض نباشد، به دو برابر حالت عادی می‌رسد.

طراحی ساده نرم‌افزار

در اکس‌پی تیم سعی می کند طراحی نرم‌افزار را تا حد ممکن ساده انجام دهد. اعضای تیم تمام انرژی خود را فقط روی نیازهای فعلی کاربران می‌گذارند و نگران نیازهای جدیدی که ممکن است در آینده مطرح شوند، نیستند. در ابتدای کار تمرکز برنامه‌نویسان تیم به انتخاب پایگاه اطلاعاتی، انتخاب معماری نرم‌افزار و ... نیست و تمام سعی تیم این است که قسمتی از نرم‌افزار را به ساده ترین راه ممکن تحویل مشتری دهد و وقتی واقعاً به تغییرات نیاز پیدا شد، می‌توانند تغییرات لازم را اعمال نمایند.

در اکس‌پی اعضای گروه می‌توانند روند تکمیلی تولید نرم‌افزار را مشاهده کنند و در جریان کار قرار گیرند. اکس‌پی روش مناسبی برای پروژه‌های کوچک است که اعضای تیم از دو تا دوازده برنامه‌نویس تشکیل شده است؛ هرچند اصولاً اکس‌پی هیچ رویه خاص و مراحل پیوسته‌ای را مشخص نکرده است. می‌توان گفت که اکس‌پی چهار مرحله اصلی دارد:

- مرحله زمانبندی پروژه
- طراحی ابتدایی
- نوشتن کدهای برنامه
- امتحان کدهای نوشته شده

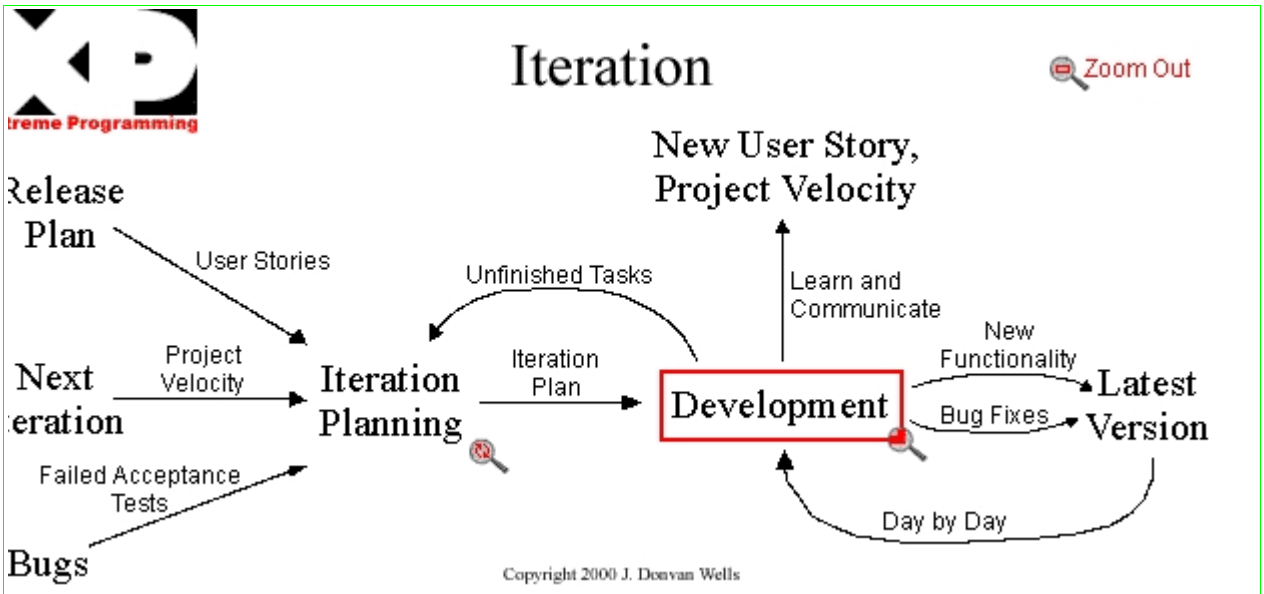
طبـق تحقیقـات انجـام شـده مشـخص شـده اسـت کـه اکس‌پـی تنهـا در پروژه‌هـای کوچـک نرم‌افـزاری می‌توانـد مفیـد باشـد و در پروژه‌هـای بـزرگ، اصـلاً موفـق نخواهـد بـود. شـاید بـه ایـن دلیـل کـه در ایـن روش مسـتندات چنـدانی بـرای نرم‌افـزار وجـود نـدارد و چنـد نفـر بیشـتر نمی‌تواننـد در مـورد قسـمتی از نرم‌افـزار اطلاعـاتی داشـته باشـند. همچنـین نرم‌افـزار تولیـد شـده بـا ایـن روش هیچ‌گونـه طراحـی سـازمان یافتـه‌ای نـدارد و ایـن امـر می‌توانـد بـرای مراحـل پـس از نصـب، یعنی تعمیرات و نگهداری سیستم، باعث بروز مشکلاتی گردد.

از جملـه مزایـای اکس‌پـی ایـن اسـت کـه از آن جـایی کـه یـک برنامـه‌نویس بـه صـورت مسـتقیم کـدهای برنامـه را چـک می‌کنـد، کیفیـت نرم‌افزارهـای تولیـدی بـالاتر مـی‌رود. همچنـین از آن جـایی کـه دو برنامـه‌نویس بـا هـم کـار می‌کننـد، آمـوزش کمتـری نیـاز اسـت و در نتیجـه هزینـه تولیـد نرم‌افـزار پـایین میآیـد، امـا ایـن روش مشـکلات خاصـی نیـز دارد. مـثلاً تصـور کنیـد اگـر در یـک گـروه یـک برنامـه‌نویس تمـایلی بـرای کـار بـا دیگـر برنامه‌نویسـان نداشـته باشـد، چـه اتفـاقی خواهد افتاد؟!

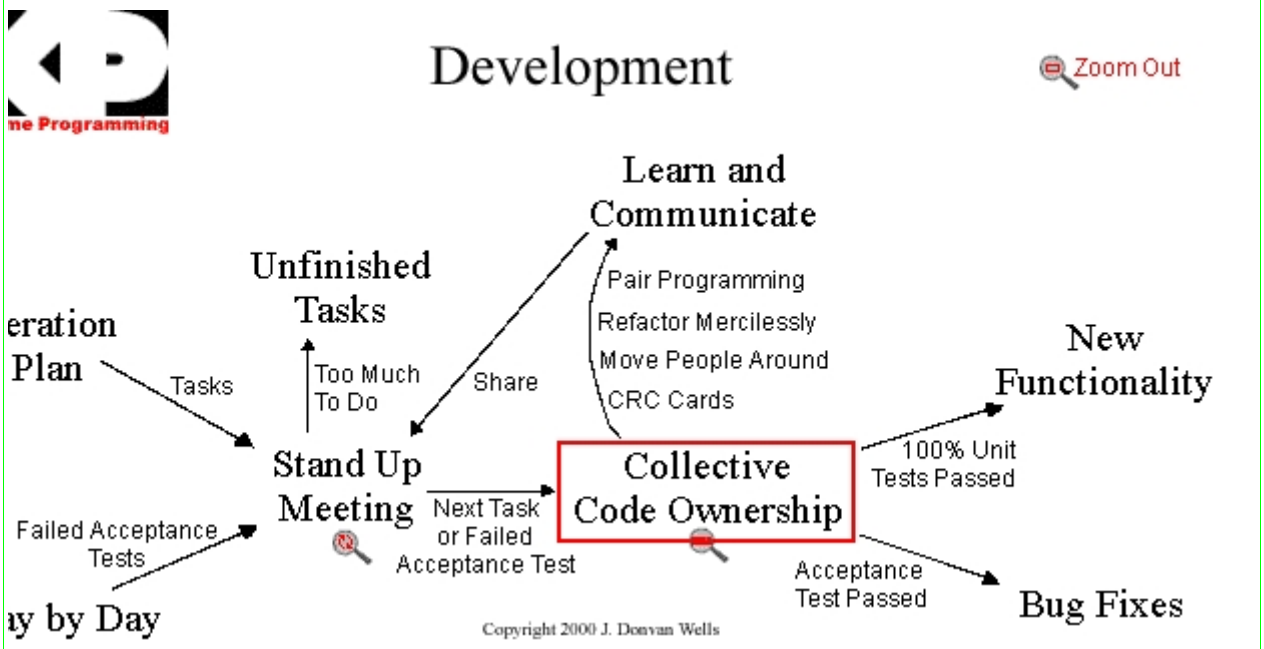
متخصصـان نرم‌افـزار پـس از تحقیقـات زیـاد روی ایـن روش بـه ایـن نتیجـه رسـیده‌اند کـه وقتـی برنامه‌نویسـی در کـدهای برنامـه بـه دنبـال اشـکال می‌گـردد، حـداکثر می‌توانـد پـانزده درصـد از اشـکالات برنامـه را پیـدا کنـد، ولـی در روش‌هـایی ماننـد اکس‌پـی کـه دو برنامـه‌نویس بـا هـم کـار می‌کننـد و یکـی از ایـن برنامه‌نویسـان کـدها را چـک می‌کنـد، تـا چهـل درصـد از اشـکالات ساختاری برنامه مشخص می‌شود.

نمودارهای مربوط به اینProcess Model

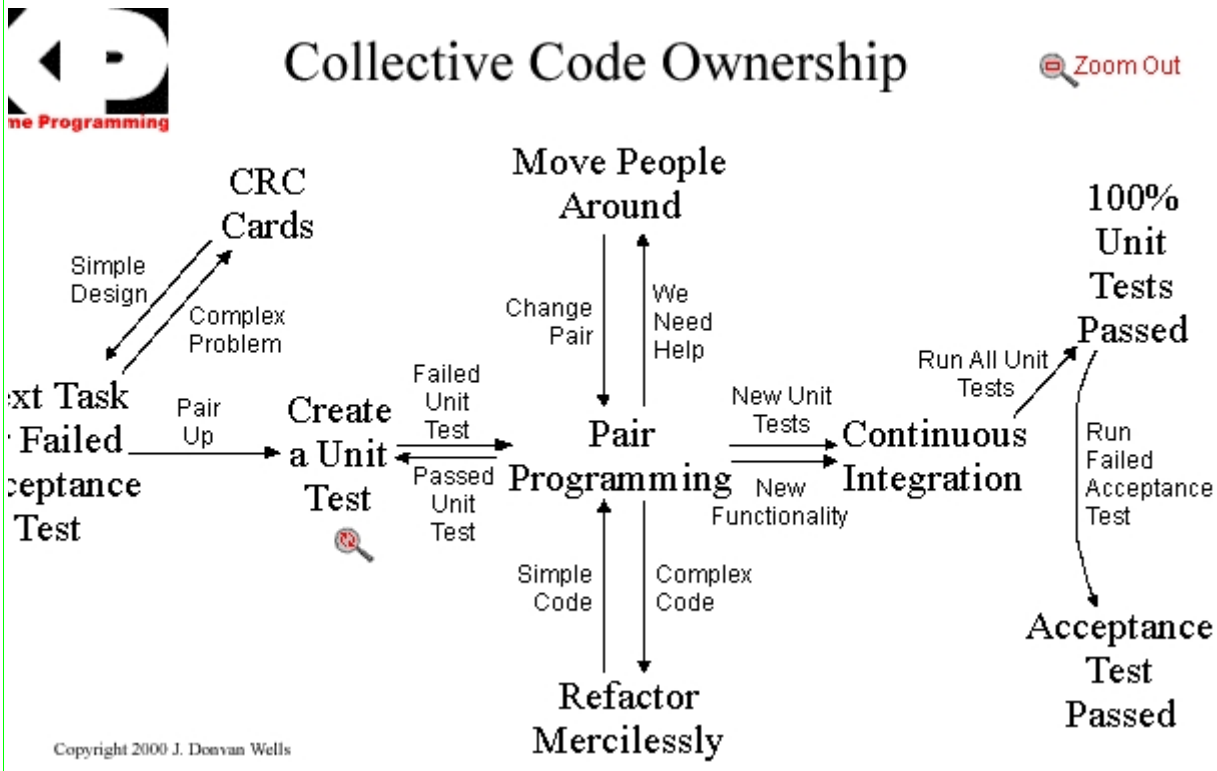
Iteration



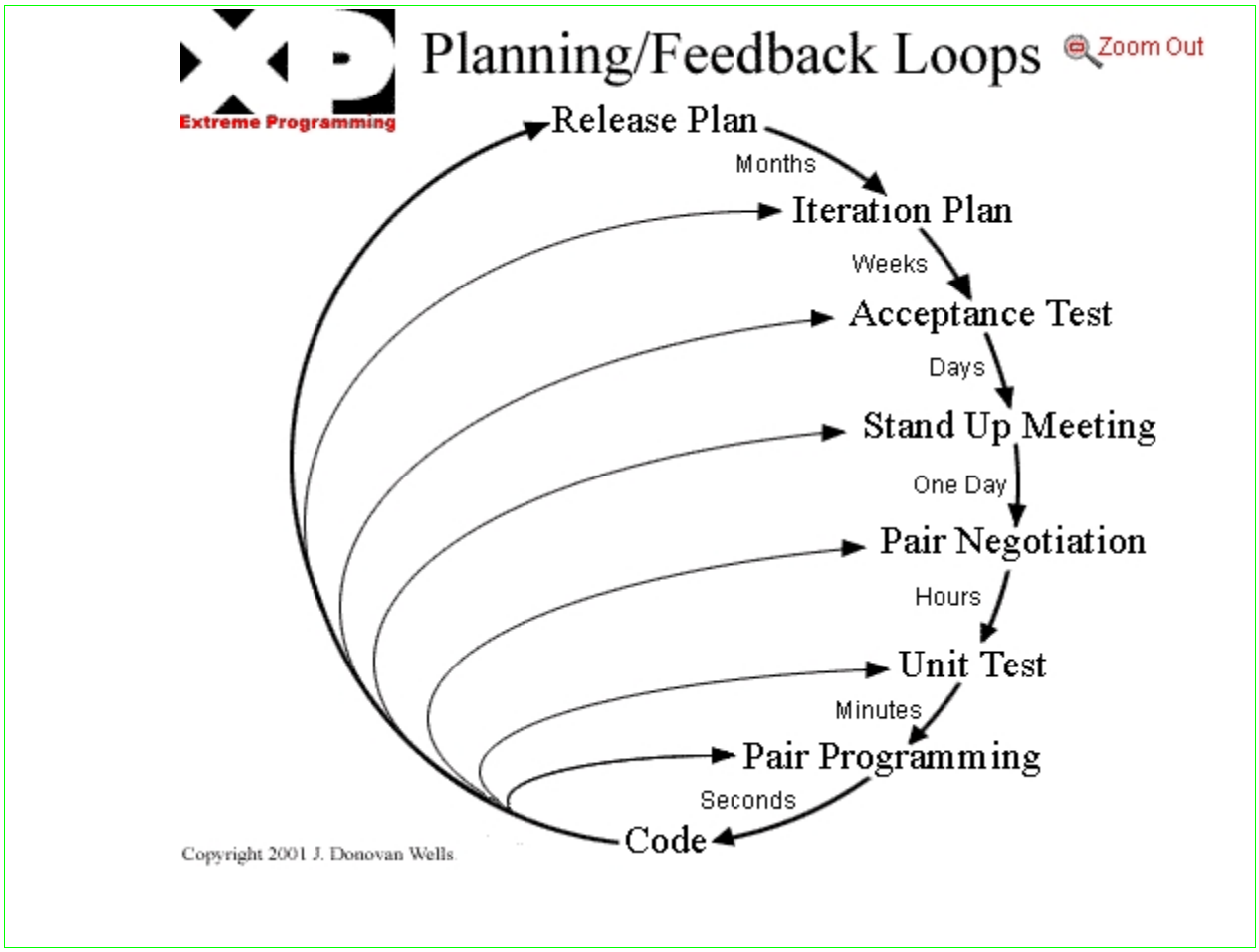
Development



Code Ownership



Planning Feedback Loops


Extreme Programming
Planning/Feedback Loops
Zoom Out
Release Plan
Months
Iteration Plan
Weeks
Acceptance Test
Days
Stand Up Meeting
One Day
Pair Negotiation
Hours
Unit Test
Minutes
Pair Programming
Seconds
Code
Copyright 2001 J. Donovan Wells