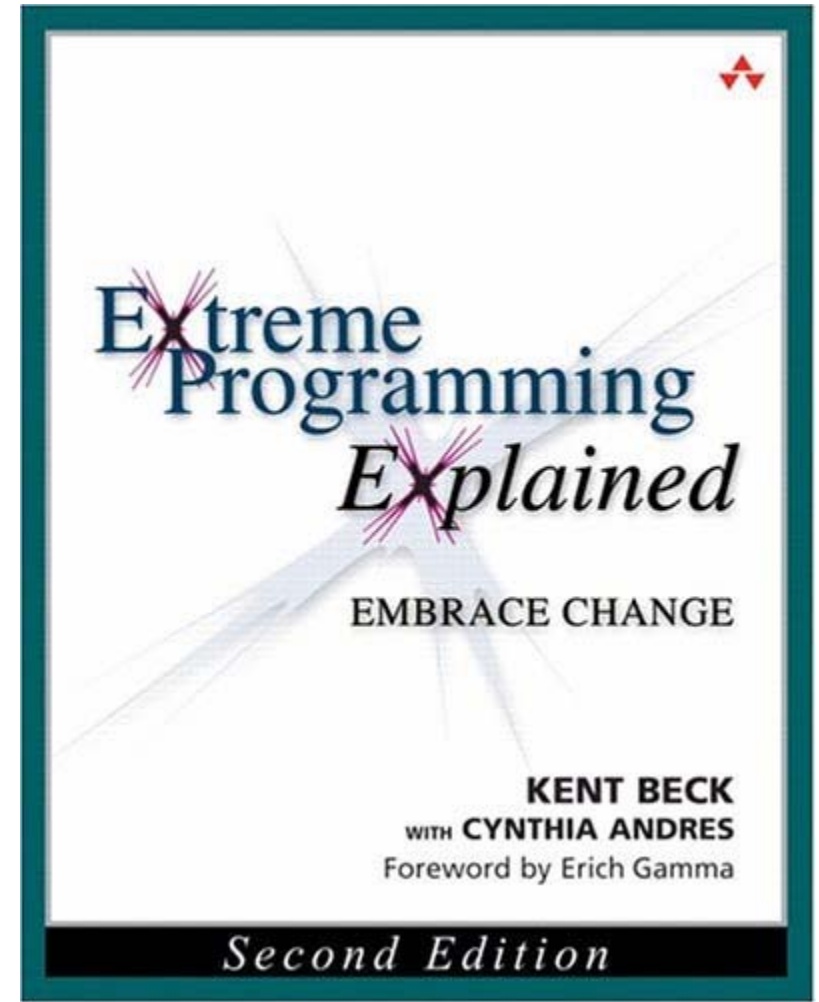


Case Study Retrospective: Kent Beck's XP Versions 1 and 2

Laurie Williams
North Carolina State University
williams@csc.ncsu.edu

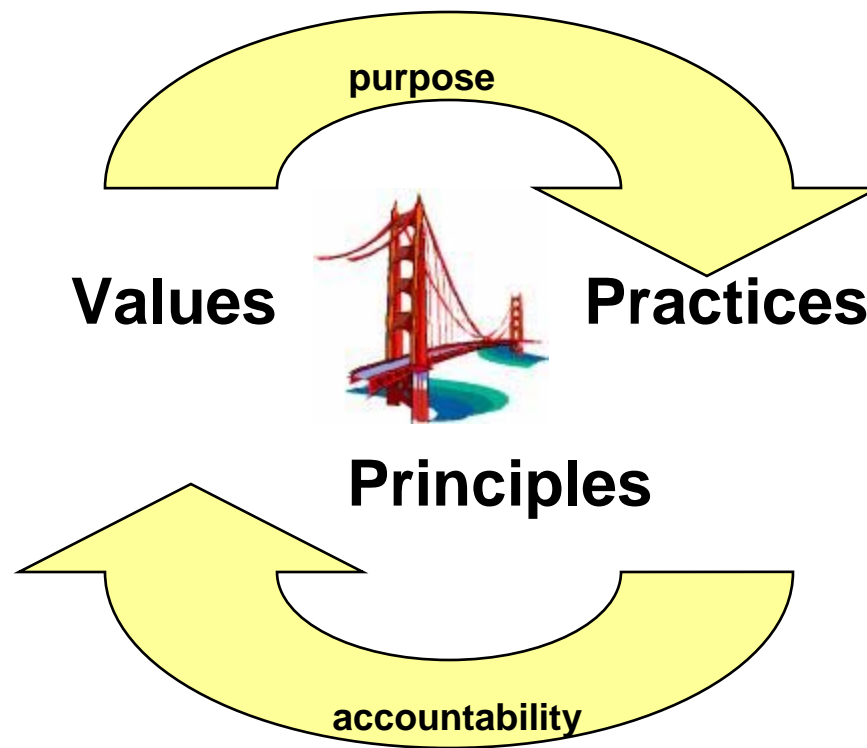


Agenda

- **Extreme Programming (XP) 2nd Edition**
 - Values
 - Principles
 - Primary Practices
 - Corollary Practices

- **Retrospective Look at Empirical Studies of XP1 Teams**
 - IBM
 - Sabre

Values, Principles, and Practices



XP2 Values

- Values are the roots of things we like and don't like in a given situation.
- Communication: Often when a problem arises, someone knows the solution but knowledge doesn't get around to person who needs it. [sustained]
- Simplicity: Making a bet that it is better to do a simple thing today and pay a little more to change it if it needs it, than to do a more complicated thing today that may never be used. [sustained]
- Feedback: No fixed direction stays valid for long. [sustained]
- Courage: Effective action in the face of fear. [sustained]
- Respect: Team members must care about each other and the project. [new]

XP2 Principles

- Principles are domain-specific guidelines.
- Humanity
- Economics
- Mutual benefit
- Self-similarity
- Improvement
- Diversity
- Reflection
- Flow
- Opportunity
- Redundancy
- Failure
- Quality
- Baby steps
- Accepted responsibility

Values

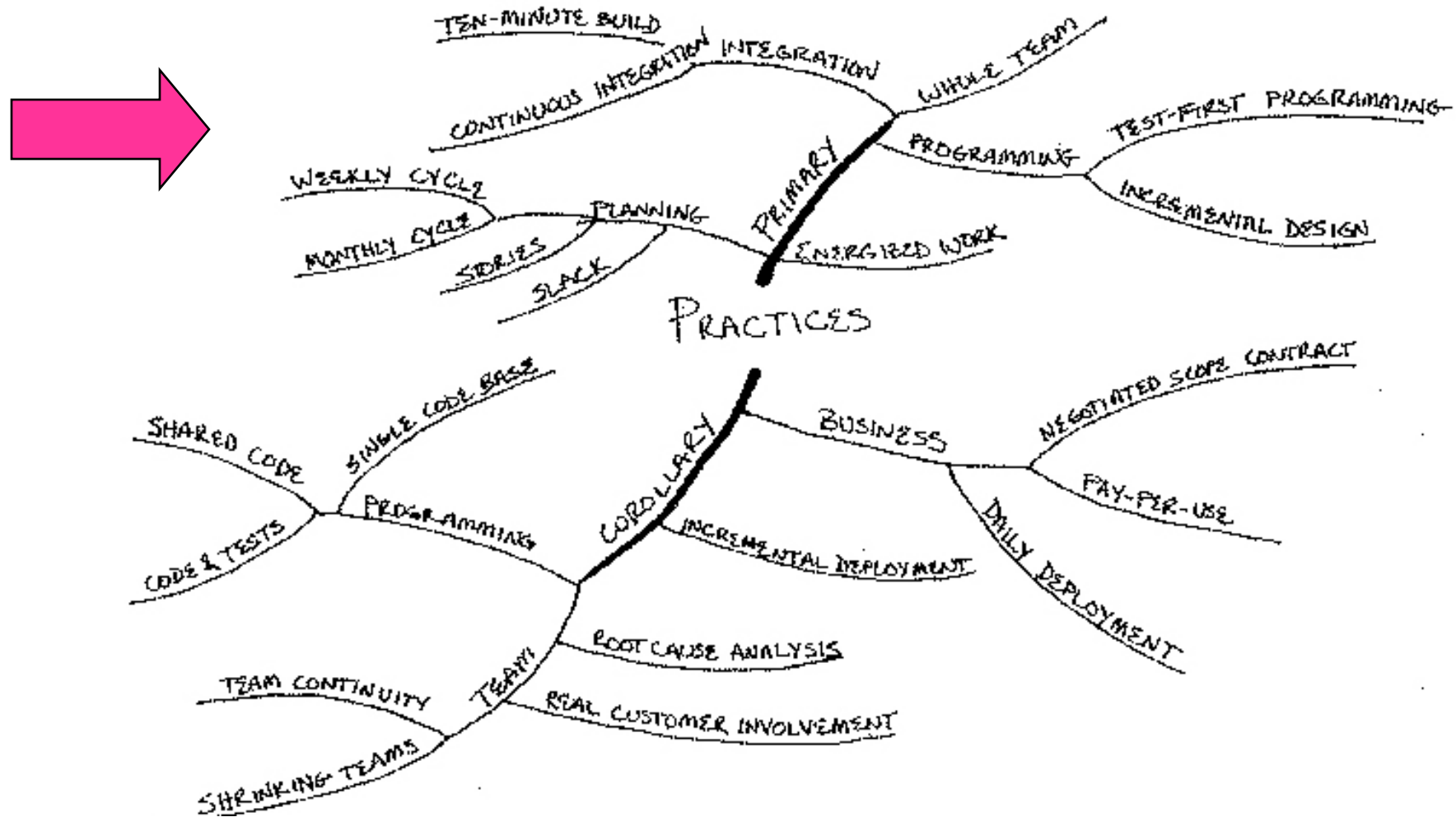


Practices

Principles

XP2 Practices: Primary

Practices are the things you do day-to-day.



Sit Together [new]



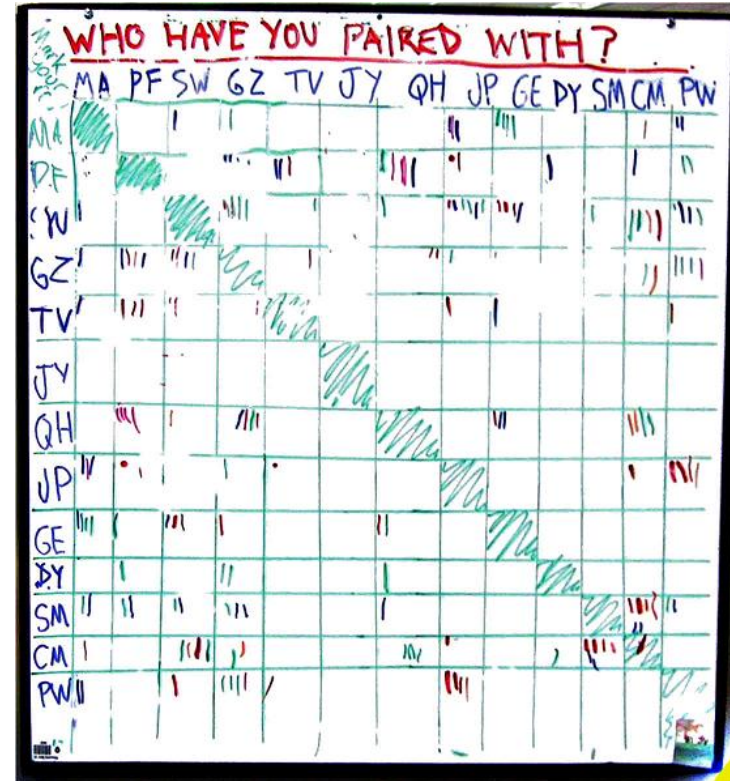
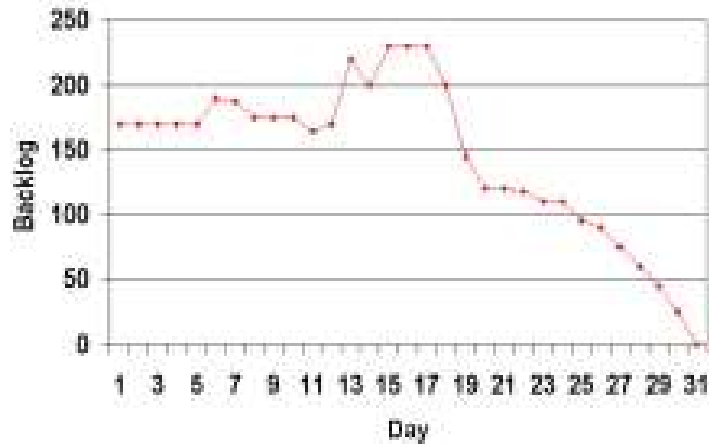
- Develop in an open space big enough for everyone.
- Have small, private spaces nearby.

Whole Team [new]

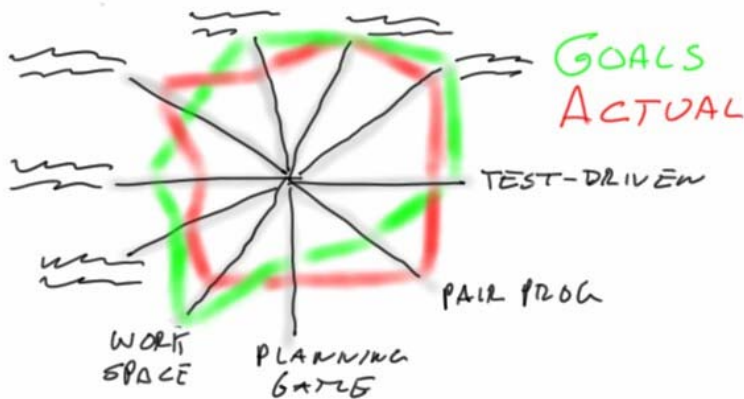


- **1st class cross-functional team**
- **Tipping Points [Malcolm Gladwell]**
 - **12:** # of people who can comfortably interact in a day
 - **150:** above this you no long recognize the faces of everyone on the team

Informative Workspace [new]



PRACTICE RADAR CHART



Energized Work [was 40-Hour Week]

- **Work only as many hours as you can be productive and only as many hours you can sustain.**
 - **Tired developers make more mistakes, which slows you down more in the long run (remove value from product).**
 - **If you mess with people's personal lives (by taking it over), in the long run the project will pay the consequences.**



Pair Programming [sustained]



- Two software engineers work on one task at one computer
- One engineer, **the driver**, has control of the keyboard and mouse and creates the implementation
- The other engineer, **the navigator**, watches the driver's implementation to identify defects and participates in on-demand brainstorming
- The roles of driver and observer are periodically rotated between the two software engineers

Stories

[was Planning Game (User Stories)]

- Customer-visible functionality

A photograph of a light blue sticky note with handwritten text in black ink. The text describes a user story for an 'Edit SR Details Screen'. It includes a title, a description of the requirement, a test case (TC), and estimated/actual time. The sticky note is placed on a white background.

Story 26
05-Oct-01

Edit SR Details Screen

Add "Cancel" - Button to undo changes and return to previous page.

TC: Check that browser returns to correct previous change

Estimated: 2h
Actual :

Weekly cycle *[was Planning Game]*

- Highest priority stories in “time boxed” weekly increments
- Caveat: see Slack practice



Quarterly Cycle *[was Small Releases]*

- Timeboxed
- As small as possible, but still delivering business value
 - No releases to 'implement the database'
- Get customer feedback early and often

Slack [new]

- In every iteration, plan some lower-priority tasks that can be dropped if you get behind – builds trust if you don't miss the “important stuff.”

Ten-Minute Build [new]

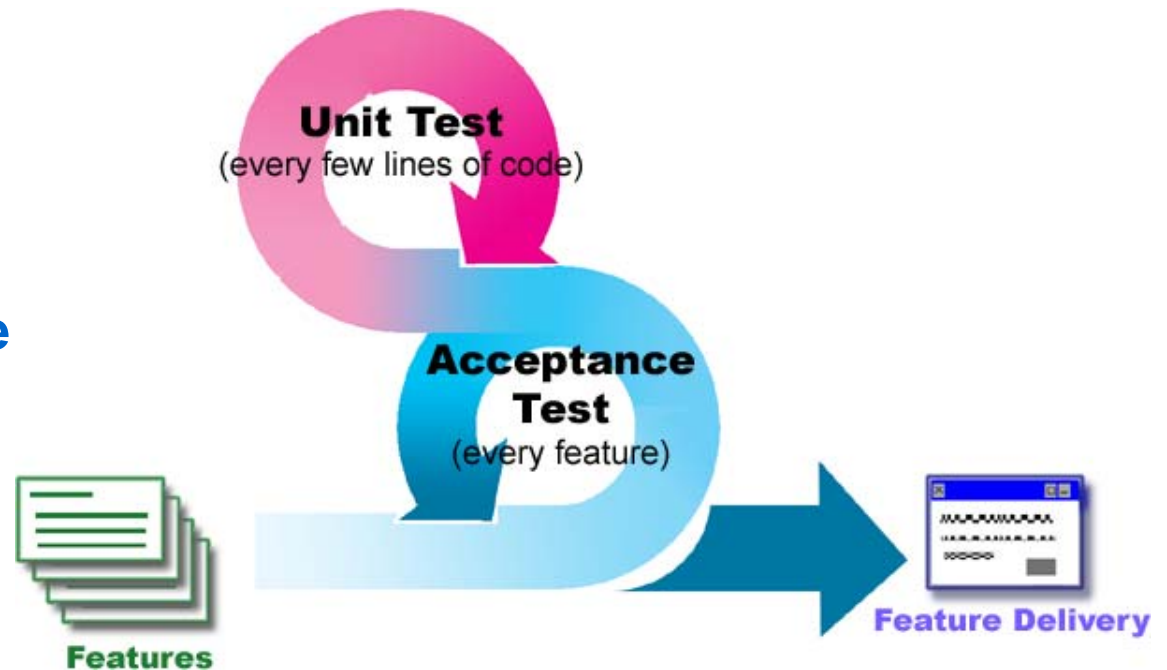
- Automatically build the entire system and run all tests in 10 minutes
- Feedback, feedback!

Continuous Integration [sustained]

- **Pair writes up unit test cases and code for a task (part of a user story)**
- **Pair unit tests code to 100%**
- **Pair integrates**
- **Pair runs ALL unit test cases to 100%**
- **Pair moves on to next task with clean slate and clear mind**
- **Should happen once or twice a day.**
- **Prevents IntegrationHell [integration could take longer than programming]**

Test-first Programming [sustained]

- **Test-Driven Development (TDD)**
 - Write tests before code
 - Tests are automated
 - Often use xUnit framework
 - Must run at 100% before proceeding
- **Acceptance Testing**
 - Written with the customer
 - Acts as “contract”
 - Measure of progress



Incremental Design

[was *Simple Design and Refactoring*]

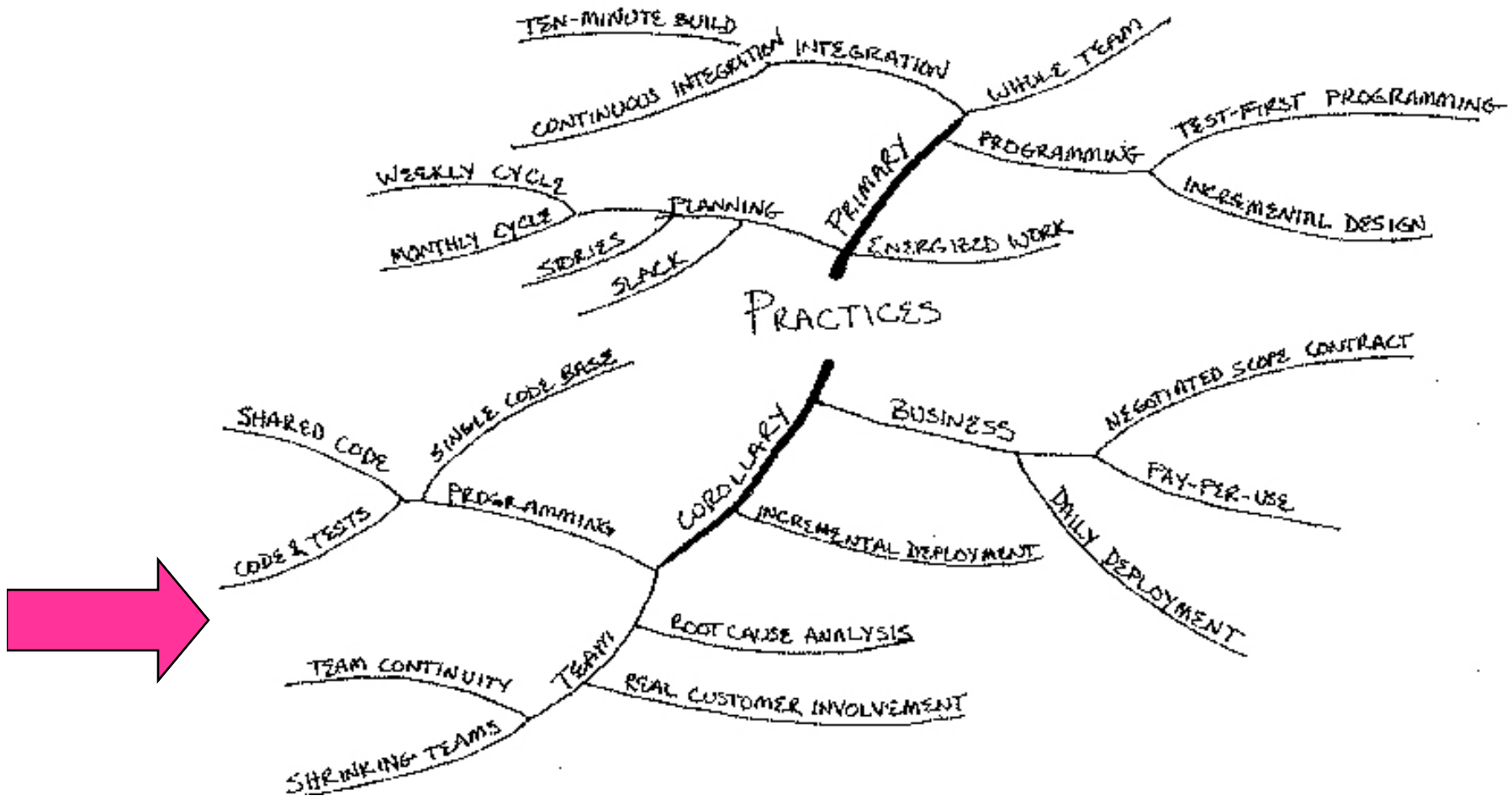
- No Big Design Up Front (BDUF)
- Knowledge-based design – the most effective design is in light of experience
- “Do The Simplest Thing That Could Possibly Work”
- “You Aren’t Gonna Need It” (YAGNI)
- Refactoring: Improve the design of existing code without changing functionality
 - Simplify code
 - Opportunity for abstraction
 - Remove duplicate code
- Relies on testing to ensure nothing breaks in the process of refactoring

XP2 Primary Practice Summary

<i>XP2 Primary Practice</i>	<i>Sustained/New/ XP1 Name</i>
Sit together	New
Whole team	New
Informative workspace	New
Energized work	40-hour week
Pair programming	Sustained
Stories	Planning game
Weekly cycle	Planning game
Quarterly cycle	Small releases
Slack	New
Ten-minute build	New
Continuous integration	Sustained
Test-first Programming	Testing
Incremental Design	Simple Design Refactoring

<i>XP1 Practice</i>	<i>Disposition</i>
Metaphor	Removed
Collective code ownership	Corollary: Shared code
On-site customer	Corollary: Real customer involvement
Coding standard	Removed

XP2 Practices: Corollary



Corollary Practices

- **Real Customer Involvement [was *On-Site Customer*].** Customer available on site to clarify stories and to make critical business decisions.
- **Incremental Deployment [new].** Gradually deploy functionality. Big deployment is high risk and can have high human and economic costs.
- **Team Continuity [new].** Keep effective teams together.
- **Shrinking Team [new].** As a team grows in capability, keep the workload constant but gradually reduce the size (e.g. with attrition).

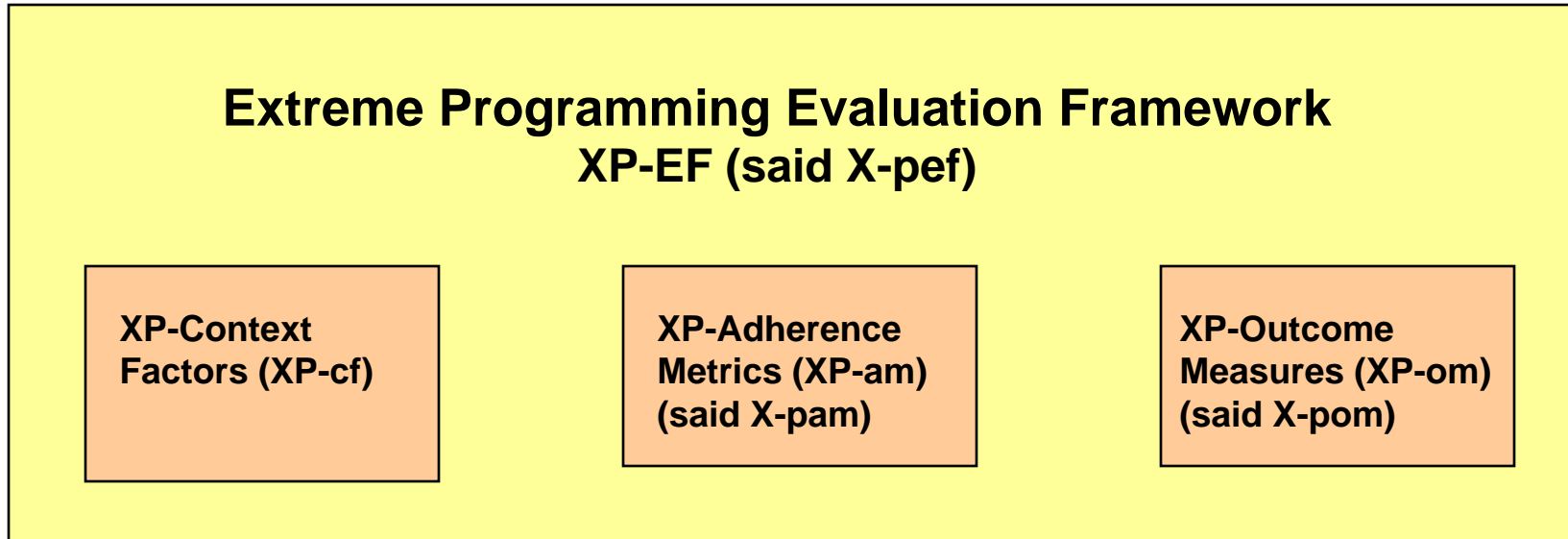
Corollary Practices (cont'd)

- **Root-Cause Analysis [new].** (1) write failing automatic system test; (2) write failing automatic unit test; (3) get each to pass; (4) examine how defect was created and not caught
- **Shared Code [was *Collective Code Ownership*].** Anyone on the team can improve any part of the system at any time. [prereq: pair programming, continuous integration; test-first programming]
- **Code & Tests [was *Simple Design*].** Maintain only the code and tests as permanent artifacts. Rely on social mechanisms to keep alive the important history of the project.
- **Single Code Base [new].** Have only one code stream.

Corollary Practices (cont'd)

- **Daily Deployment [new].** Put new code into production every night.
- **Negotiated Scope Contract [new].** Fix time, cost, and quality but call for on-going negotiation of precise scope.
- **Pay-per-use [new].** Charge for every time the system is used.

Extreme Programming Examination



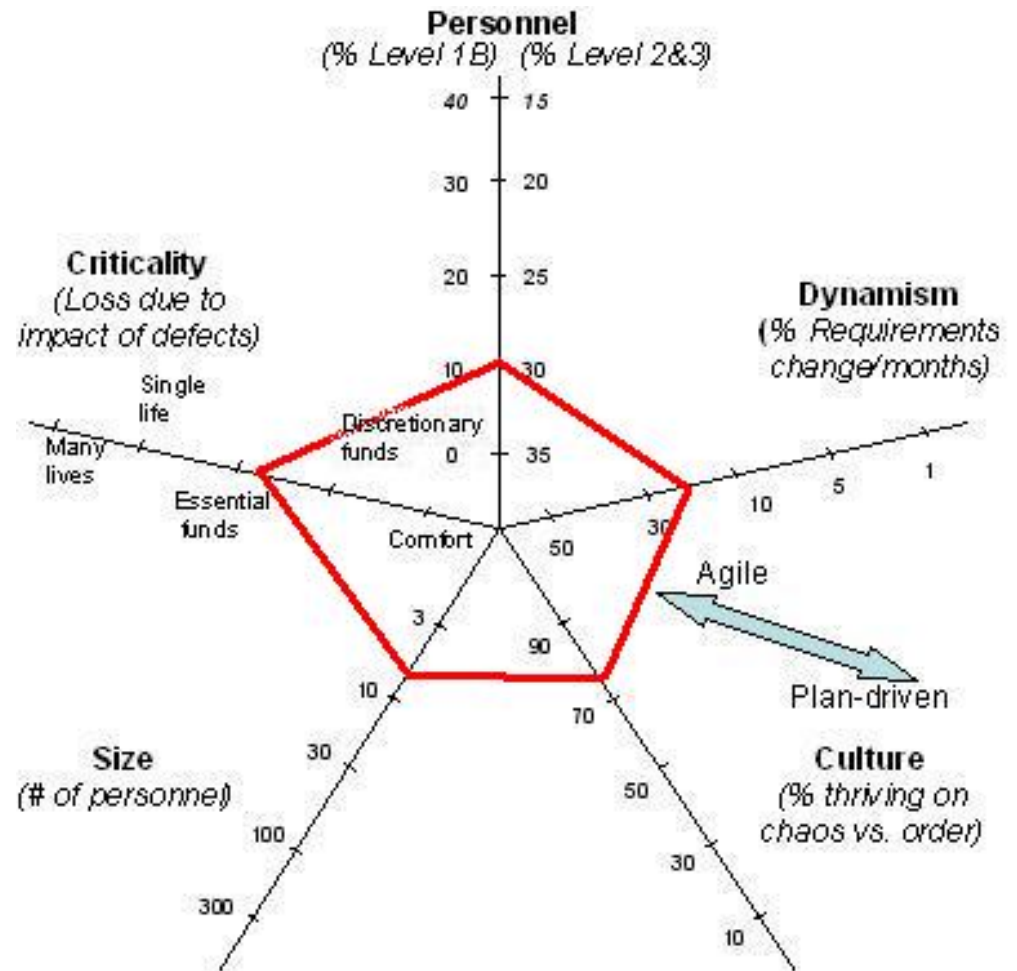
- **Reusable framework for reporting:**

- **the extent to which an organization has adopted XP practices;
and**
- **the result of this adoption**

IBM: XP-Context Factors (XP-cf)

- Small team (7-10)
- Co-located
- Web development (toolkit)
- Supplier and customer distributed (US and overseas)

- Examined one release “old” (low XP) to the next “new” (more XP)

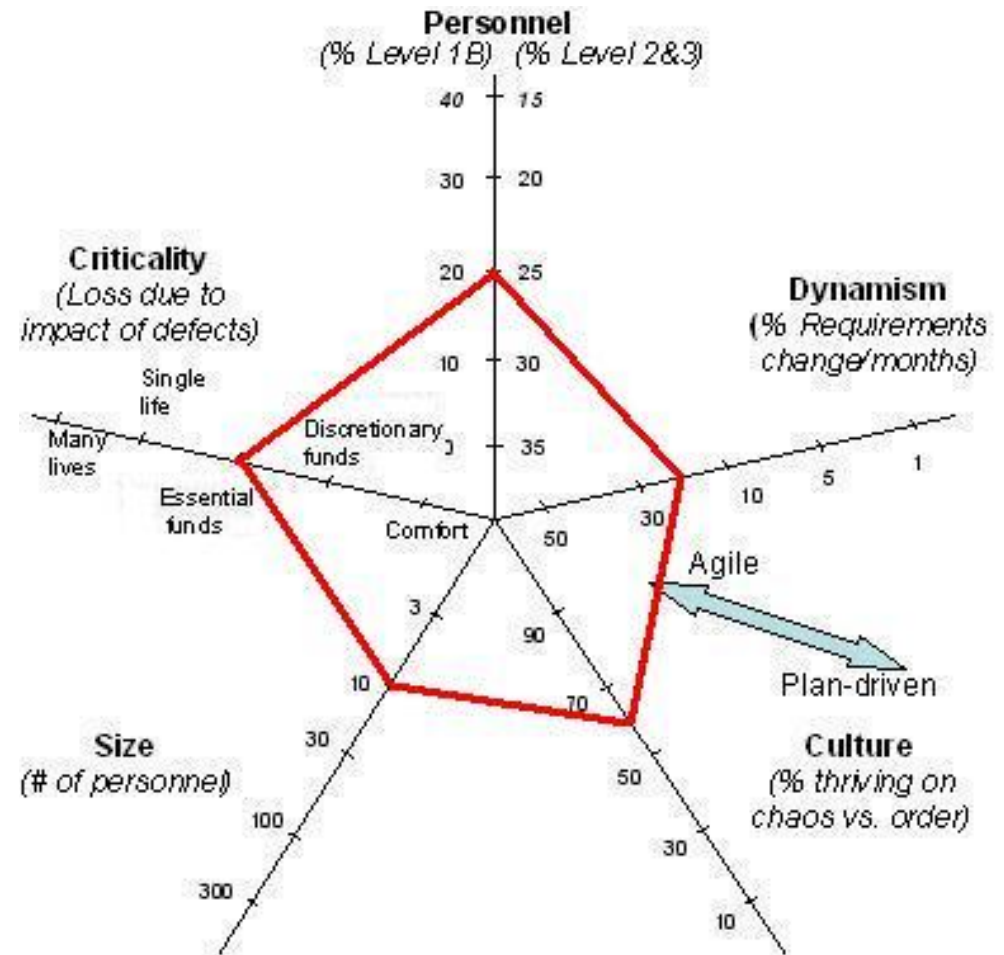


IBM: XP-Outcome Measures (XP-om)

XP Outcome Measures	Old	New
Response to Customer Change (Ratio (user stories in + out) /total)	NA	0.23
Pre-release Quality (test defects/KLOEC of code)	1.0	0.50
Post-release Quality (released defects/KLOEC of code)	1.0	0.61
Productivity (stories / PM)	1.0	1.34
Relative KLOEC / PM	1.0	1.70
Putnam Product. Parameter	1.0	1.92
Customer Satisfaction	NA	High (qualitative)
Morale (via survey)	1.0	1.11

Sabre: XP-Context Factors (XP-cf)

- Small team (6-10)
- Co-located
- Scriptable GUI environment
- Customer remote, multinational, several time zones
- Examined third release “old” (low XP) to the ninth release “new” (sustained XP)



Sabre: XP-Outcome Measures (XP-om)

XP Outcome Measures	Old	New
Response to Customer Change (Ratio (user stories in + out) /total)	NA	N/A
Pre-release Quality (test defects/KLOEC of code)	1.0	0.25
Post-release Quality (released defects/KLOEC of code)	1.0	0.70
Productivity (stories / PM)	N/A	N/A
Relative KLOEC / PM	1.0	1.46
Putnam Product. Parameter	1.0	2.89
Customer Satisfaction	NA	High (anecdotal)
Morale (via survey)	N/A	68.1%

Summary

Two characteristically-agile teams:

<i>When used by teams operating within the specified context, the use of a specified subset of XP practices leads to an improvement in . . .</i>		
Hypothesis	IBM Case study evidence?	Sabre case study evidence?
<i>pre-release quality</i>	Yes	Yes
<i>post-release quality</i>	Yes	Yes
<i>programmer productivity</i>	Yes	Yes
<i>customer satisfaction</i>	Yes	N/A
<i>team morale</i>	Yes	N/A

XP2 Primary Practices

<i>XP2 Primary Practice</i>	<i>IBM</i>	<i>SABRE-A</i>
Sit together	No – adjoining cubes	Yes
Whole team	No	Yes (customer rep sitting at times)
Informative workspace	No	Yes (many big visible charts)
Energized work	Sustainable pace	Sustainable pace
Pair programming	50% anecdotal	50% anecdotal
Stories	Yes	Yes
Weekly cycle	Yes	10 day
Quarterly cycle	5 months	Yes
Slack	Not likely	Not likely
Ten-minute build	No	No (hours to build)
Continuous integration	Nightly or more	Daily
Test-first Programming	Progress in unit testing No automated acceptance testing	Progress in unit testing Some automated acceptance testing
Incremental Design	SDUF Limited refactoring	No design doc Limited refactoring

XP1 Primary Practice Rejects

<i>XP1 Practice</i>	<i>IBM</i>	<i>SABRE-A</i>
Metaphor	No	System of names
Collective code ownership	Yes	Yes
On-site customer	No (remote, responsive to email)	On-site marketing rep (1/2 time; email)
Coding standard	Yes	Naming standard

Conclusions

- **XP2 has 13 primary practices**
 - Can do individually, work best together
- **XP2 has 11 corollary practices**
 - Best to start using these once have a core set of primary practices
- **XP2 seems more “reasonable” than XP1**
- **Two small, co-located, **successful** XP1 teams were studied**
 - **IBM team used:**
 - » ~8 of the 13 XP2 primary practices
 - » 2 of 4 XP1 rejected XP1 primary practices
 - **Sabre team used:**
 - » ~12 of 13 XP2 primary practices
 - » ~3 of 4 XP1 rejected XP1 primary practices