

Industrial Case Studies of Extreme Programming Teams

Sabre / Airline Solutions /



TEKELEC



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



Real Engineers – Real Projects – Real Impact

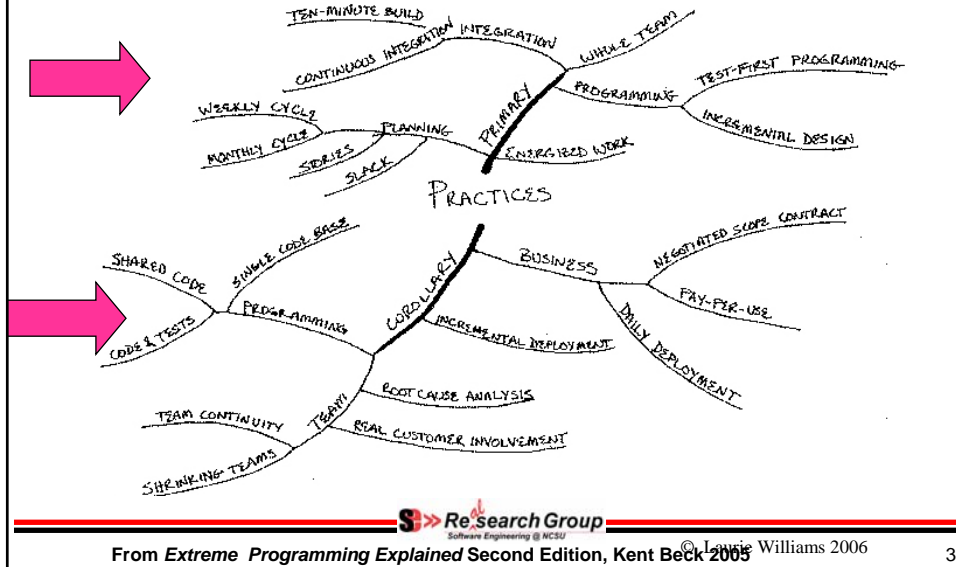
Agenda

- Brief Overview: Extreme Programming (XP) 2nd Edition (XP2)
- Empirical Research Design of XP1 Studies
- Empirical Studies of XP1 Teams
 - IBM
 - Two Sabre
 - Tekelec
- Summary and Research Challenges

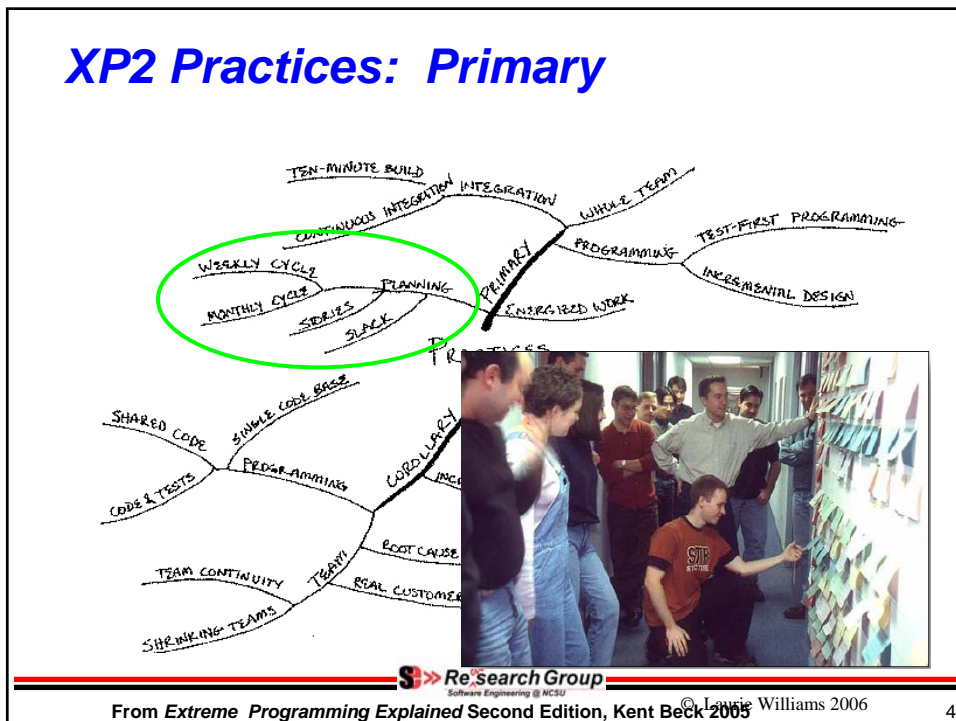


© Laurie Williams 2006

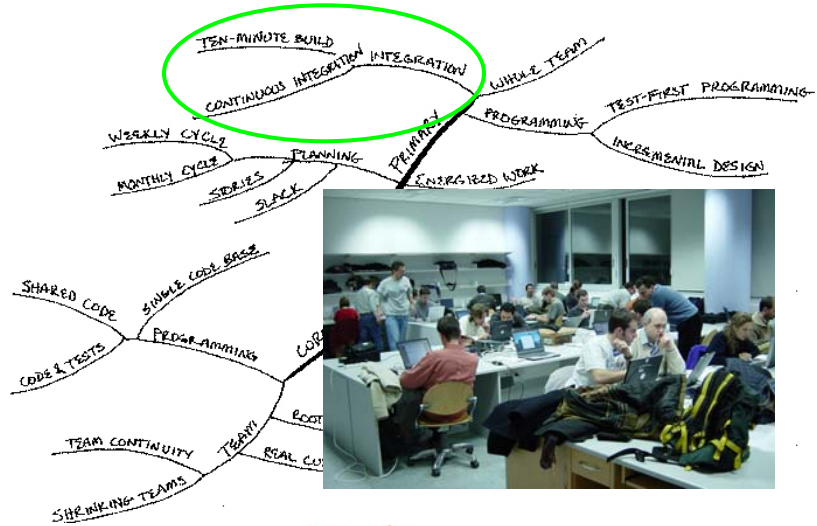
XP2 Practices



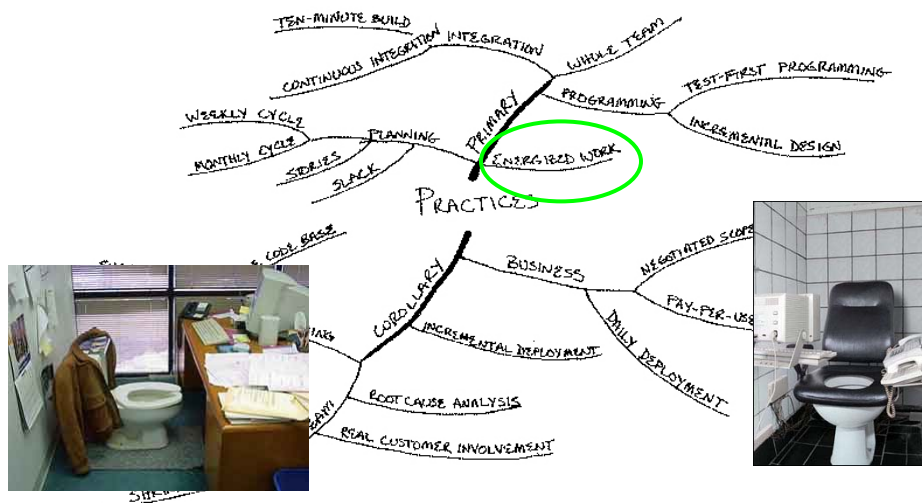
XP2 Practices: Primary



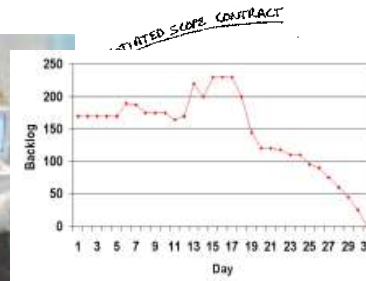
XP2 Practices: Primary



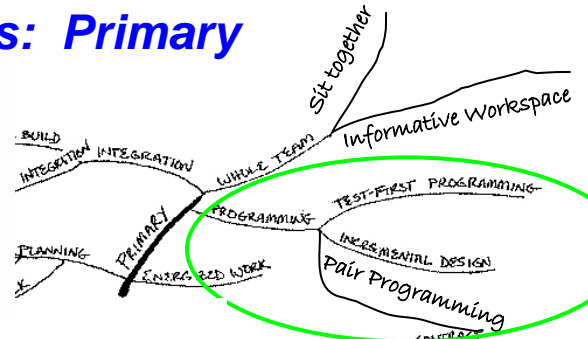
XP2 Practices: Primary



XP2 Practices: Primary



XP2 Practices: Primary

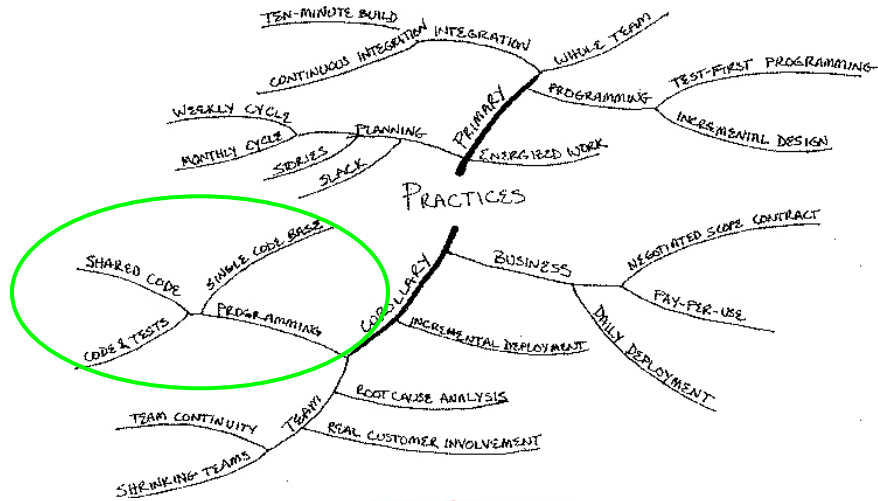


Unit Test
(every few lines of code)

Acceptance Test
(every feature)



XP2 Practices: Corollary



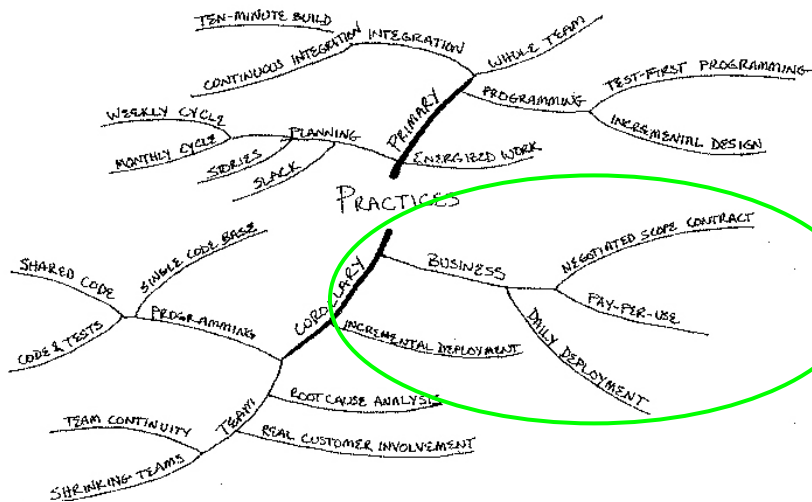
Re:search Group
Software Engineering @ NCSU

From Extreme Programming Explained Second Edition, Kent Beck 2005

© Louie Williams 2006

9

XP2 Practices: Corollary



Re:search Group
Software Engineering @ NCSU

From Extreme Programming Explained Second Edition, Kent Beck 2005

© Louie Williams 2006

10

XP2 Practices: Corollary



Re:search Group
Software Engineering @ NCSU
From Extreme Programming Explained Second Edition, Kent Beck 2005 © Laurie Williams 2006 11

XP2 Primary Practice Summary

| XP2 Primary Practice | Sustained/New/ XP1 Name |
|------------------------|----------------------------|
| 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 |

| XP1 Practice | Disposition |
|---------------------------|--------------------------------------|
| Metaphor | Removed |
| Collective code ownership | Corollary: Shared code |
| On-site customer | Corollary: Real customer involvement |
| Coding standard | Removed |

Re:search Group
Software Engineering @ NCSU

© Laurie Williams 2006

12

Agenda

- Brief Overview: Extreme Programming (XP) 2nd Edition (XP2)
- Empirical Research Design of XP1 Studies
- Empirical Studies of XP1 Teams
 - IBM
 - Two Sabre
 - Tekelec
- Summary and Challenges

XP: Goal – Question – Metric

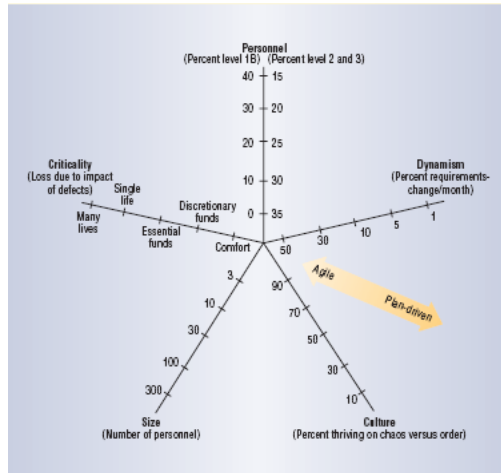
Goal: To build laws and evolve theories about whether the business-related results of a team change when XP practices are used.

Example Law: The use of XP increases customer satisfaction.

Example Theory: Because of the continuous communication between the development team and the customer, the product is more likely to be what the customer *actually* wants, rather than what the customer *initially stated* he/she wanted.

- Q1: Does the pre-release quality change when a team uses XP practices?
- Q2: Does the post-release quality change when a team uses XP practices?
- Q3: Does programmer productivity change when a team uses XP practices?
- Q4: Does customer satisfaction change when a team uses XP practices?
- Q5: Does team morale change when a team uses XP practices?
- Metrics . . . Will be discussed in detail.

Software Development Process Selection

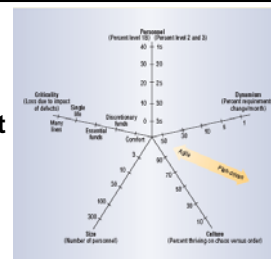


Process Selection – GQM

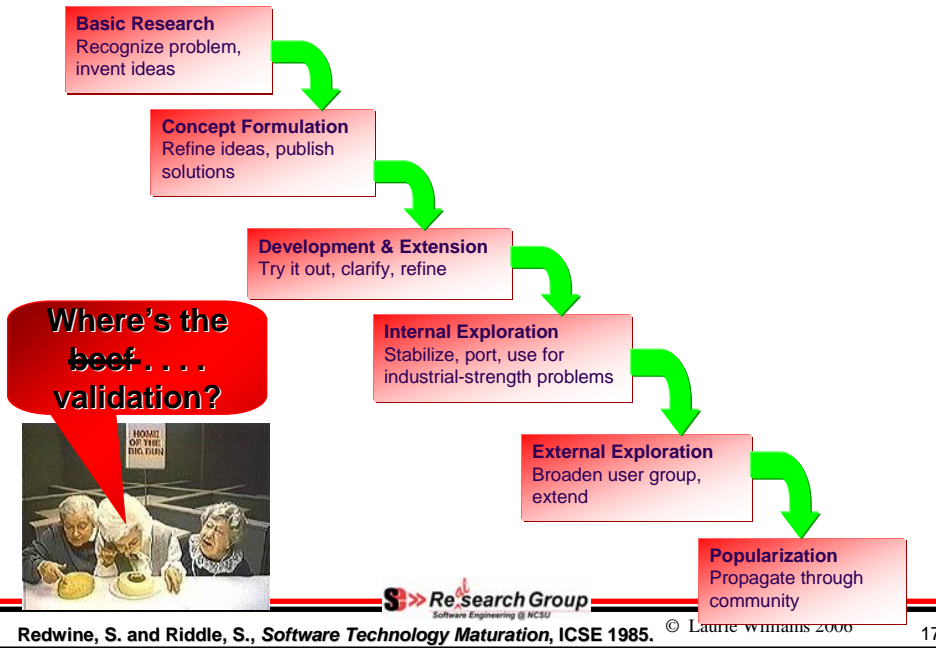
- **Goal:** to build laws and evolve theories about systematic guidelines for software development process selection.

Questions:

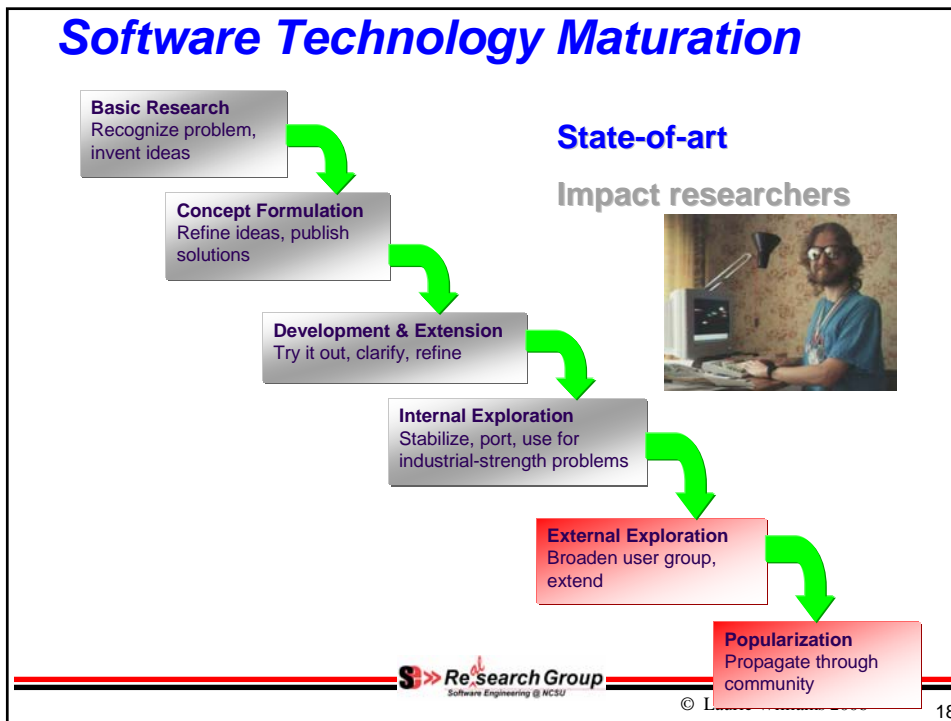
- Are the most important factors in choosing a plan-driven or agile methodology the following: personnel, dynamism, culture, size, and criticality? ... And co-location?
- **Personnel:** Should lower-skill teams should use plan-driven methodologies; higher skilled teams can use agile methodologies?
- **Dynamism:** Should projects with lower requirements volatility should use plan-driven methodologies; projects with higher requirements volatility should use agile methodologies?
- **Culture:** Should teams comprised of engineers who like order should use plan-driven methodologies; teams comprised of engineers who thrive on chaos should use agile methodologies?
- **Size:** Should large teams should use plan-driven methodologies; small teams should use agile methodologies?
- **Criticality:** Should reliability-critical projects should use plan-driven methodologies; projects with minimal implications of defects should use agile methodologies?



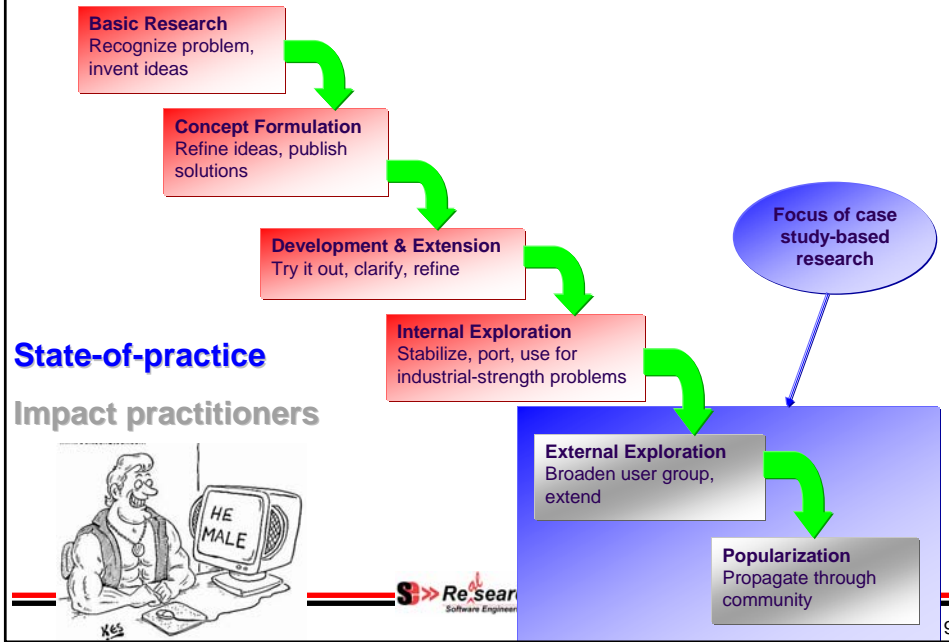
Software Technology Maturation



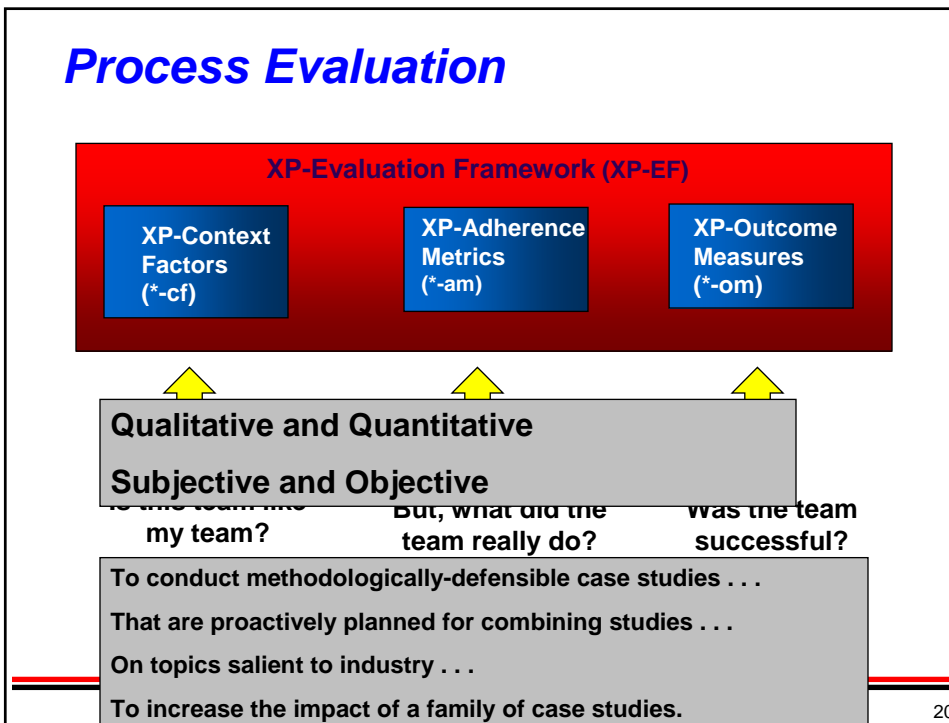
Software Technology Maturation



Software Technology Maturation



Process Evaluation



Documenting How/What to Measure

5.1. Quality Measures

- 1) **Pre-release Quality (test defects/KLOEC):** This metric reflects quality exposed during test before it is release to a customer such as is done by an external testing group within an organization. The metric is a *surrogate* measure of quality [KIT96], it is also a measure of testing efficiency.

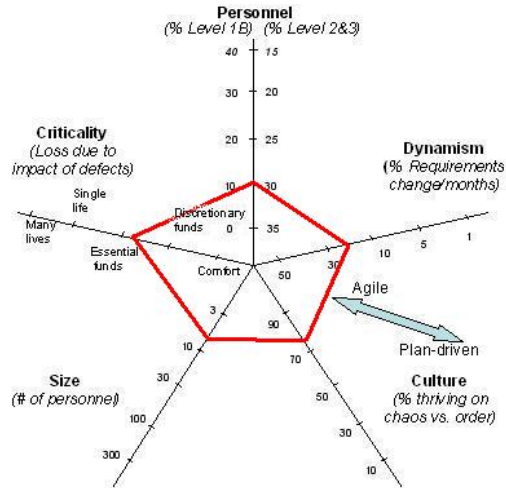
| | |
|------------|--|
| Count | Count the number of defects found in the new and changed code. These defects are found during final system-level testing before the product is released to a customer. This testing can be done by the development team or by an external testing team in the development organization. Count the lines of code of the Delta set. |
| Exclude | Do not count defects discovered or reported during the release but not in the new or changed lines of code. Do not count those defects found by the developers during unit testing Do not count bugs in reused binary code libraries from other teams not a part of the study. Do not count bugs that were reported but were duplicates or irreproducible. Do not count bugs in test or sample code not shipped to customers. Do not count bugs found by the compiler or during pairing or inspections. Do not count bugs injected after the release (those belong in Post-release Quality). |
| Additional | Include information on the severity of the defects found as an indication of whether defects were critical or minor. Also, it is advisable to include information about the testing effort exhibited by the testing team. Testing effort can be measured in terms of person-months, number of scenarios tested, etc. Also document when during development external testing takes place, e.g. throughout development, just prior to release, etc. This information is very important for comparing pre-release defect density. A lower pre-release defect density could be due to improved product quality or a non-thorough testing effort. |

Agenda

- **Brief Overview: Extreme Programming (XP) 2nd Edition (XP2)**
- **Empirical Research Design of XP1 Studies**
- **Empirical Studies of XP1 Teams**
 - IBM
 - Two Sabre
 - Tekelec
- **Summary and Challenges**

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-Adherence Metrics (XP-am)

- Subjective: Shodan Survey
 - Example survey at: http://agile.csc.ncsu.edu/survey/shodan_survey.html
 - Old 56%
 - New 72%
- Objective Metrics

| XP-am Metric | Practice | Old | New |
|--|------------------|-----------|----------|
| Automated test class per user story | Testing | 0.11 | 0.45 |
| Test coverage (statement) | Testing | 30% | 46% |
| Unit test runs per person day | Testing | 14% | 11% |
| Test LOC/Source LOC | Testing | 0.26 | 0.42 |
| Accept test execute | Testing | Manual | Manual |
| Did customers run your acceptance tests? | Testing | No | No |
| Pairing Frequency | Pair Programming | <5% | 48% |
| Release Length | Short Release | 10 months | 5 months |
| Iteration Length | Short Release | Weekly | Weekly |

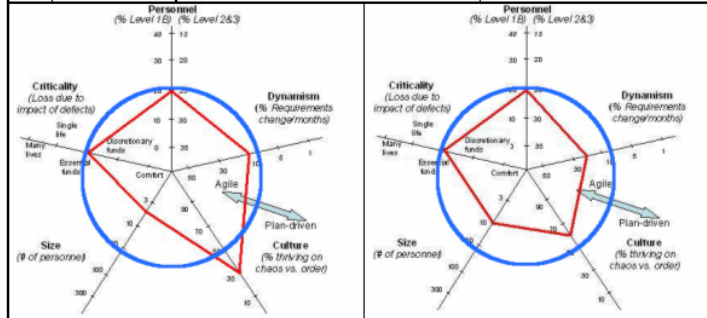
IBM: XP-Outcome Measures (XP-om)

- Normalized values

| 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-A: XP Context Factors (XP-cf)

| Context factor | Old release | New release |
|------------------------------|--|---------------------|
| Software Dev. Meth. | Waterfall with some XP practices | XP |
| Team size | 6 | 10 |
| Personnel turnover | 67% | 10% |
| Team location | Co-located | |
| Customer location | approx. 5 overseas | approx. 11 overseas |
| Language | Java, C++ | Java |
| Physical layout | Semi-private cubicles | Open lab |
| Constraints | Date constrained, scope constrained, semi-resource constrained | |
| Domain | Scriptable GUI environment | |
| Person months | 108 | 14.7 |
| Component KLOEC ³ | 133.8k | 193.4k |



³ Thousands of lines of executable code

Sabre-A: XP-Adherence Metrics (XP-am)

- **Subjective: Shodan Survey**
 - 76.7% (new)
- **Objective Metrics**

| XP-am Metric | Practice | Old | New |
|--|------------------|-----------|------------|
| Automated test class per new/changed class | Testing | 0.036 | 0.572 |
| Test coverage (statement) | Testing | N/A | 32.9% |
| Unit test runs per person day | Testing | 0 | 1.0 |
| Test LOC/Source LOC | Testing | 0.054 | 0.296 |
| Accept test execute | Testing | Manual | Manual |
| Did customers run your acceptance tests? | Testing | No | No |
| Pairing Frequency | Pair Programming | <0% | 50% |
| Release Length | Short Release | 18 months | 3.5 months |
| Iteration Length | Short Release | -- | 10 days |

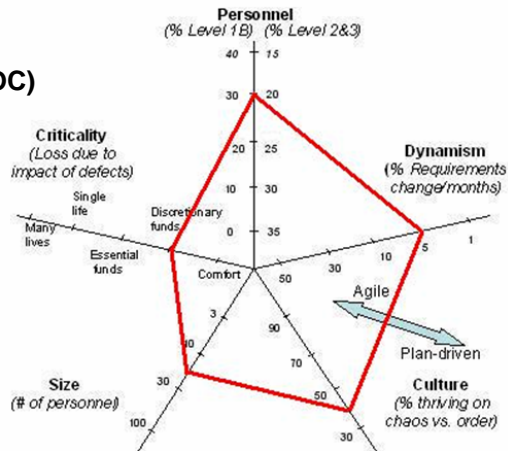
Sabre-A: XP-Outcome Measures (XP-om)

- **Normalized values**

| 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 | 0.35 |
| 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% |

Sabre-P: XP Context Factors (XP-cf)

- Medium sized team (15)
- Co-located
- Large web application (1M LOC)
- Customers domestic & overseas
- Examined 13th release of the product; 20 months after starting XP



Sabre-P: XP-Adherence Metrics (XP-am)

- Subjective: Shodan Survey
– 70.2
- Objective Metrics

| XP-am Metric | Practice | New |
|--|------------------|----------|
| Automated test class per new/changed class | Testing | 0.0225 |
| Test coverage (statement) | Testing | 7.7% |
| Unit test runs per person day | Testing | 0.4 |
| Test LOC/Source LOC | Testing | 0.296 |
| Pair programming | Pair programming | 70% |
| Release Length | Short Release | 3 months |
| Iteration Length | Short Release | 10 days |

Sabre-P: XP-Outcome Measures (XP-om)

| XP Outcome Measures | Bangalore SPIN Benchmarking group | Capers Jones |
|----------------------------|-----------------------------------|--------------|
| Pre-release defect density | Similar | Lower |
| Total defect density | Lower | Lower |
| Productivity | Similar | Higher |

Tekelec: XP Context Factors (XP-cf)

- **Small team (4-7; 2 during maintenance phase)**
- **Geographically distributed**
 - Contractors in Czech Republic for US development organization (Tekelec)
- **Simulator for a telecommunications signal transfer point system (train new customers)**
- **Considerable amount of requirements volatility**

Tekelec: XP-Adherence Metrics (XP-am)

| XP-am Metric | Practice | New |
|--|------------------|------------------------------------|
| Automated test class per new/changed class | Testing | 1.0 |
| Test coverage (statement) | Testing | N/A |
| Unit test runs per person day | Testing | 1/day for all; 1/hour for quickset |
| Test LOC/Source LOC | Testing | 0.91 |
| Pair programming | Pair programming | 77.5% |
| Release Length | Short Release | 4 months |
| Iteration Length | Short Release | 10 days |

Tekelec: XP-Outcome Measures (XP-om)

| Outcome measure | F-15 project |
|---|---|
| Pre-release Quality (test defects/KLOEC) | N/A |
| Post-release Quality (post-release defects/KLOEC) | 1.62 defects/KLOEC [Lower than industry standards] |
| Customer Satisfaction (interview) | Capability – Neutral Reliability – Satisfied Communication – Very Satisfied |
| Productivity | 1.22 KLOEC/PM [Lower than industry standards] 2.32 KLOEC/PM (including test code) [on par with industry standards] |

Tekelec: Conjectures





- *Conjecture 1: In a globally-distributed XP team, a well-defined customer authority is essential for effective decision making and a clear requirements statement.*
- *Conjecture 2: In a globally-distributed XP team, having a key member of one team physically located with the other team can provide an essential two-way communication conduit.*
- *Conjecture 3: In a globally-distributed XP team, prompt responses to asynchronous queries positively impact development commitment and confidence and create a focused development environment.*
- *Conjecture 4: In a globally-distributed XP team, providing the team with continuous access to process and product information (e.g. XPlanner) can help to improve process control and plan effectiveness.*

Agenda

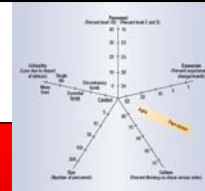
- **Brief Overview: Extreme Programming (XP) 2nd Edition (XP2)**
- **Empirical Research Design of XP1 Studies**
- **Empirical Studies of XP1 Teams**
 - IBM
 - Two Sabre
 - Tekelec
- **Summary and Challenges**

Empirical Studies of XP Teams

The use of a specified subset of XP practices leads to an improvement in . . .

| | IBM | Sabre-A | Sabre-P | Tekelec |
|-------------------------|---|---|--|---|
| Hypothesis |  |  |  |  |
| | Small, Co-located | Small, Co-located | Medium, Co-located | Small, Distributed |
| pre-release quality | Yes | Yes | Similar-Yes | N/A |
| post-release quality | Yes | Yes | Yes | Yes |
| programmer productivity | Yes | Yes | Similar-Higher | No |
| customer satisfaction | Yes | N/A | N/A | Neutral-satisfied |
| team morale | Yes | N/A | N/A | N/A |
| Evaluation Framework | 1.3 | 1.2 | 1.4 | 1.4 |

Software Process Selection



*-Evaluation Framework

-Context Factors (-cf)

-Adherence Metrics (-am)

-Outcome Measures (-om)

● XP-EF → XP-cf; XP-am; XP-om







● RUP-EF → RUP-cf; RUP-am; RUP-om



● TSP-EF → TSP-cf; TSP-am; TSP-om



● XP2-EF → XP2-cf; XP2-am; XP2-om

References

- **IBM Case Study:** Williams, L., Krebs, W., Layman, L., Antón, A., *Toward a Framework for Evaluating Extreme Programming*, Empirical Assessment in Software Engineering (EASE) 2004, Edinburgh, Scotland, pp. 11-20.
- **Sabre-A Case Study:** Layman, L., Williams, L., Cunningham, L., *Exploring Extreme Programming in Context: An Industrial Case Study*, Agile Development Conference 2004, Salt Lake City, p. 32-41.
- **Sabre-P Case Study:** Layman, L., Williams, L., Cunningham, L., *Motivations and Measurements in an Agile Case Study*, Journal of System Architecture, to appear.
- **Tekelec Case Study:** Layman, L., Williams, L., Damian, D., Bures, H., *Essential Communication Practices for Extreme Programming in a Global Software Development Team*, Information and Software Technology, to appear.
- Available from:
<http://collaboration.csc.ncsu.edu/laurie/publications.html>