

CSE 333 – Software Engineering

Pranta Sarker
Lecturer
Dept. of CSE, NEUB

Lecture 08

Outline

- Agile development
- Agility
- Agility principles
- Agile process concept
- Extreme programming (XP)

Agile Development

- **Philosophy encourages**
 - Customer satisfaction
 - Early incremental delivery of software
 - Small, highly motivated **project teams**
 - Informal methods
 - Minimal SE work products
 - Overall development simplicity
- **Development guidelines encourages**
 - Delivery over analysis and design
 - Active and continuous communication between developers and customers.

Agile Development

- Responsible personnel
 - Software developers and other project stakeholders
 - Managers, customers, and users
- Fundamental steps
 - Communication
 - Planning
 - Modeling
 - Construction
 - Deployment

Agile Development

- Work product
 - *Software increment*, which is delivered to the customers on the appropriate commitment time
- When have I done?
 - If the agile team agrees that the process works, and the team produces deliverable software increments that satisfy the customer, you've done it right

Agility concept

- The changes
 - In the software being built
 - To team members
 - Changes cause of new technology
 - Changes on all kinds that have an impact on the product or the project
- Respond to changes

Agility Principles

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Businesspeople and developers must work together.

Agility Principles

5. Build projects around **motivated individuals**. Give them the environment and support they need and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is **face-to-face conversation**.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

Agility Principles

- 9. Continuous attention to **technical excellence and good design** enhances agility.
- 10. **Simplicity**—the art of maximizing the amount of work not done—is essential.
- 11. The best **architectures, requirements, and designs** emerge from self-organizing teams.
- 12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Agile Process

- **Unpredictability**
 - Which software requirement will persist
 - Which prioritizes to be changed by customer
 - Analysis, design, construction, and testing
- **An agile process must be adaptable**
 - Incrementally
 - Customer feedback
- **Incremental development strategy**
 - Software increments
 - » Executable prototype or portions of an operational system

Agile Process – Extreme Programming (XP)

- The most widely used to agile software development.
- XP values
 - **Communication:** between developers and other stakeholders could be close, yet informal (verbal) collaboration for the establishment of effective **metaphors**.
 - **Simplicity:** Design and coding. Keeping simple design leads an easy implementation. If the design must be improved, it can be **refactored**.
 - **Feedback:**
 - Implemented software itself
 - Customer and
 - Software team

Extreme Programming - XP Values

- XP values
 - Courage
 - Each agile team must be disciplined and design for the present in lieu of the future so that they can mitigate the future changes.
 - Respect
 - The agile team inculcates respect among its members between other stakeholders and team members, and indirectly for the software itself.

Extreme Programming - XP Process

- Planning

- Begins with *listening* – a requirements gathering activity by enabling technical members of the XP team.
- Creation of a set of “stories” or user story that describe required output, features, and functionality.
- Story is written by the customer and placed on an index card.
- Customer assigns a value (i.e., a priority). Members of XP team then then assess ach story and estimate a cost.
- If the story is estimated to require more than 3 weeks, the customer is asked to split the story into smaller stories and the assignment of value and cost occurs again.

Extreme Programming - XP Process

- Planning

- Customers and developers work together to decide how to group stories into the next release (the next software increment) to be developed by the XP team.
- Once a basic *commitment* is made for a release, the XP team orders the stories that will be developed in one of three ways:
 - All stories will be implemented immediately.
 - The stories with highest priority will be implemented first.
 - The riskiest stories will be moved up in the schedule and implemented first.

Extreme Programming - XP Process

- Planning

- Project velocity

- After the first release (software increment) of the software the XP team computes the project velocity defines **how many customer story they have implemented during the first release.**
 - It might be used to –
 - Help estimate delivery dates and schedule for subsequent releases and
 - Determine whether an overcommitment has been made.

Extreme Programming - XP Process

- Design
 - KIS (keep it simple)
 - Encourage Class Responsibility Collaborator (CRC) cards as an effective mechanism for thinking about the software in an object-oriented context.
 - If a **difficult design problem** is encountered as part of the design of a story, XP recommends the immediate **creation of an operational prototype** of that portion of the design. Called a *spike solution*, the design prototype is implemented and evaluated.
 - Also, XP encourages the refactoring.

Extreme Programming - XP Process

- Coding

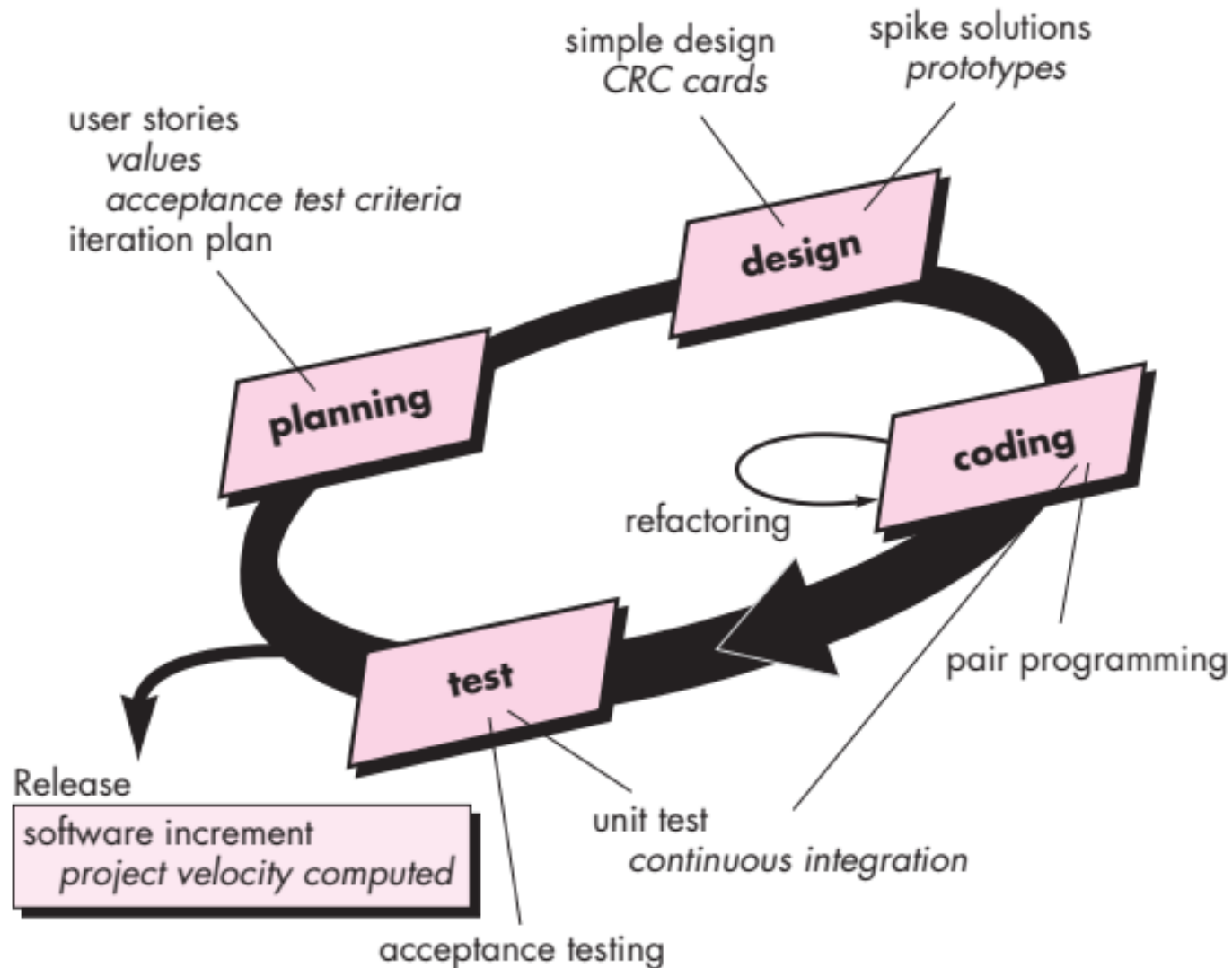
- After stories are developed and preliminary design work is done, the team does *not* move to code, but rather **develops a series of unit tests** that will exercise each of the stories that is to be included in the current release (software increment).
- A key concept during the coding activity is *pair programming*. XP recommends that two people work together at one computer workstation to create code for a story, which leads **real time problem solving and quality assurance**.

Extreme Programming - XP Process

- Testing

- Creation of unit testing before coding.
- XP *acceptance tests*, also called *customer tests*, are specified by the customer and focus on overall system features and functionality that are visible and reviewable by the customer.
- Acceptance tests are derived from user stories that have been implemented as part of a software release.

Extreme Programming - XP Process



Code refactoring

- Refactoring is the process of changing a software system in such a way that it does not alter the external behavior of the code yet improves the internal structure.
- It is a disciplined way to clean up code [and modify/simplify the internal design] that minimizes the chances of introducing bugs. In essence, when you refactor, you are improving the design of the code after it has been written.

Reference book

Software Engineering

A Practitioner's Approach

By

Roger S. Pressman

Edition: 7th

Thank you!!