Kiểm tra Phần mềm

Bài 10

Agile – Extreme Testing

Lecturer: **Nguyễn Ngọc Tú**
Email: Tu.NguyenNgoc@hoasen.edu.vn
Web: sites.google.com/site/WebKiemTraPhanMem/
#AdTekDev #ICoTek

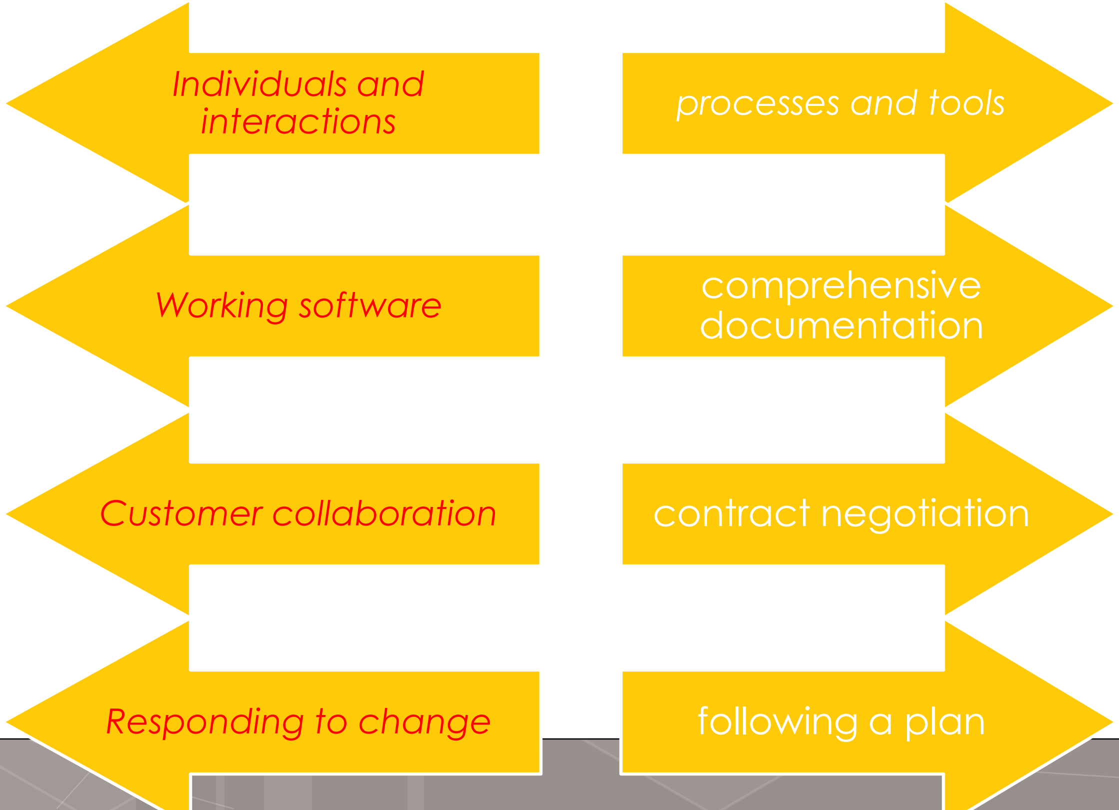
Content

- Features of Agile Development
- Agile Testing
- Extreme Programming and Testing
  - Extreme Testing: The Concepts
  - Extreme Unit Testing
  - Extreme Acceptance Testing

Features of Agile Development

- We are uncovering better ways of developing software by doing it and helping others do it.

(*Manifesto of Agile Software Development*)



Individuals and interactions

processes and tools

Working software

comprehensive documentation

Customer collaboration

contract negotiation

Responding to change

following a plan

Features of Agile Development

- **Methodologies**
  - Agile Modeling
  - Agile Unified Process
  - Dynamic Systems Development Method
  - Essential Unified Process (EssUP)
  - Extreme Programming
  - Feature Driven Development
  - Open Unified Process
  - Scrum
  - Velocity Tracking

Agile Testing

- Agile testing is a form of **collaborative testing**, in that everyone is involved in the process through design, implementation, and execution of the test plan.
- **Customers** are involved in defining **acceptance tests** by defining use cases and program attributes.
- Developers collaborate with testers to build test harnesses that can test functionality automatically.
- Agile testing requires that everyone be engaged in the test process, which requires a lot of communication and collaboration.

Extreme Programming and Testing

- facilitates the creation of quality programs in short time frames
- relies heavily on unit and acceptance testing

- to create the unit (module) and acceptance tests first, then your code base. – Extreme Testing (XT)

Extreme Programming Basics

- Implementing simple designs.
- Communicating between developers and customers.
- Continually testing the code base.
- Refactoring, to accommodate specification changes.
- Seeking customer feedback

Extreme Programming Basics

- 4 groups – 12 core practices
  - Listening to the customer and other programmers.
  - Collaborating with the customer to develop the application’s specification and test cases.
  - Coding with a programming partner.
  - Testing, and retesting, the code base.

Extreme Programming Basics

- **12 core practices**
  - Planning and requirements
  - Small, incremental releases
  - System metaphors
  - Simple designs
  - Continuous testing
  - Refactoring
  - Pair programming
  - Collective ownership of the code
  - Continuous integration
  - Forty-hour workweek
  - On-site customer presence
  - Coding standards

Extreme Programming Basics

- XP Testing
  - Continuous testing is central to the success of a XP-based effort
  - supports refactoring efforts used to optimize and streamline the code base

your customers' confidence in their investment soars because they *know the code base passes unit tests every day.*

Ex. XP Project Flow

1. *Programmers meet with the customer to determine the product requirements and build user stories.*
2. *Programmers meet without the customer to divide the requirements into independent tasks and estimate the time to complete each task.*
3. *Programmers present the customer with the task list and with time estimates, and ask them to generate a priority list of features.*
4. *The programming team assigns tasks to pairs of programmers, based on their skill sets.*
5. *Each pair creates unit tests for their programming task using the application's specification.*

Ex. XP Project Flow

6. *Each pair works on their task with the goal of creating a code base that passes the unit tests.*
7. *Each pair fixes, then retests their code until all unit tests have passed.*
8. *All pairs gather every day to integrate their code bases.*
9. *The team releases a preproduction version of the application.*
10. *Customers run acceptance tests and either approve the application or produce a report identifying the bugs/deficiencies.*
11. *Upon successful acceptance tests, programmers release a version into production.*
12. *Programmers update time estimates based on latest experience.*

Extreme Testing: The Concepts

- XT mandates creating tests before coding begins, not after

- XT Unit Testing
  - All code modules must have unit tests before coding begins,
  - All code modules must pass unit tests before being released into acceptance testing.

Extreme Unit Testing

- important benefits
  - You gain confidence that your code will meet its specification and requirements.
  - You express the end result before you start coding.
  - You better understand the application’s specification and requirements.
  - You may implement simple designs initially and confidently refactor the code later to improve performance, without worrying about breaking the specification

Extreme Acceptance Testing

- determines whether the application meets other requirements, such as functionality and usability.
- You and the customer create the acceptance tests during the design/planning phases.
- Customers provide the unbiased verification that the application meets their needs.

Q/A ?!