ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ
ΠΡΟΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ
ΜΑΘΗΜΑ: ΤΕΧΝΟΛΟΓΙΑ ΛΟΓΙΣΜΙΚΟΥ
AGILE DEVELOPMENT / MSF

Διδάσκων:
Γ. Χαραλαμπίδης, Επ. Καθηγητής
MSF Agile Essentials
Agenda

Setting the Stage
  • Agile Methodologies

Microsoft Solutions Framework
  • The Framework

MSF Agile
  • The Methodology
Setting the Stage
Defining Agile Development

- Better communication
  - Customer, your team, other teams
  - ‘executable’ requirements, less unneeded documentation
- Deal with constant change
  - Make it a customer right
  - Shorter iterations
  - Simpler and stricter requirements management
  - Have practices that help mitigate the risk of change
    - Unit testing, continuous integration etc..
- Improve and adapt the way the team works
- Give value quick, by priority
  - Incremental development, keep it simple
  - Customer has more responsibility
Agile Manifesto

- **Individuals and interactions** over processes and tools
- **Working software** over comprehensive documentation
- **Customer collaboration** over contract negotiation
- **Responding to change** over following a plan

Source: http://www.agilemanifesto.org/
What are we trying to solve?

- High failure rate of projects
  - Time, Budget, Quality, Cancellation
- Find out about problems much earlier in the process
- Compensate for not predicting the future
  - Will the client change their mind?
  - Do we really know how much it will cost to build?
  - Did we make the design good enough?
  - Did we take all the requirements?
  - Will our team stay the same throughout?
  - Do we understand the task fully?
Dealing with unpredictability

Short Iterations

Keep it simple

A better priority mechanism

Team based estimation
Some example methodologies

XP, Scrum
**Scrum**

- Short iterations
- Customer prioritization
- Team estimation
- Clear and simple artifacts
  - Reports, documentation
- Daily meeting (Scrum)
- Adjustable process

**eXtreme Programming**
Overview of Scrum

- Unit Testing & TDD
- Pair Programming
- CI
- others...

- Write Stories
- Priority selection
- Writes customer Tests
- Runs tests at end of each iteration
The road to MSF Agile…

- Take the previous…
- Add Roles…
- Add Work Streams…
- Add Microsoft Best Practices…

*This is MSF Agile!*
Introduction to MSF Core
Microsoft Solutions Framework

Established in 1991, v4 Core formally made public October 2006

Solution delivery framework for creating software

Related to MOF, Microsoft Operational Framework

Which concentrates on the management of IT infrastructure
Key components of MSF 4.0 Core

Team Model
• Program Management, Product Management, Architecture, Development...

Governance
• Tracks: Envision, Plan, Build, Stabilize, Deploy
• Go/No Go project checkpoints

Iterations
• Potentially shippable product increase

Daily Builds
• The heart of the project

Readyness Management
• FIRO based model for empowering the team

Risk Management
• Practices for attacking project risks
Elements of an Iterative Approach
Agenda

- Agile Software Development with MSF
- Team Model
- Personas/Scenarios
- The “Agile” Pattern
- Architecture
- Testing
What is MSF for Agile Software Development?

MSF for Agile Software Development is an **iterative, scenario-driven, context-based** software development **process** for building .NET, Web, Web Service, and other object-oriented applications.  

*It guides the team through the whole project*
The process template defines key aspects of how the team project is managed. The process template may include work item types, work products, reports, queries, and process guidance for your team project.

**Which process template should be used to create the team project?**

- MSF for Agile Software Development
- MSF for Agile Software Development
- MSF for CMMI Process Improvement

Choose the MSF for Agile Software Development process for projects with short lifecycles and results-oriented teams who can work without lots of intermediate documentation. MSF for Agile Software Development is an iterative, scenario-driven development process for building .NET, Web, Web Service, and other object-oriented applications that directly incorporates practices for handling quality of service requirements such as performance and security, utilizes a context-driven approach (context-based) to determine how to operate the project, explicitly calls out project risk as a success criteria for the optimal delivery of software, and incorporates all of the traditional software development roles such as the business analyst, project manager, architect, developer, tester, and release manager. This MSF process is a flexible guidance framework that helps create an adaptive system for software development. This agile methodology anticipates the need to adapt to change, and focuses on people as the most important component to the success of a project. It also emphasizes the delivery of working software and promotes customer validation as key success measures. Choose MSF for CMMI Process Improvement over MSF for Agile Software Development, if your organization is undertaking a broad quality assurance and process improvement initiative or...
Activities in MSF

- Composed of 14 basic work streams
  - Basic activity building blocks of MSF
  - A work stream is an activity that is composed of other activities
  - Contains 70 activities (not including work streams)
  - Most work streams are performed by a single role
Work Streams

- Capture Product Vision
- Create a Scenario (use case - like)
- Create a Quality of Service Requirement
- Plan an Iteration
- Create Solution Architecture
- Implement a Development Task
- Build a Product
- Test a Scenario
- Test a Quality of Service Requirement
- Fix a Bug
- Close a Bug
- Release a Product
- Guide Project
What’s different from other agile processes?

- Context Driven (Work Streams)
- Recognizes all the -itties (QoS)
- Roles
- Adds Microsoft Best Practices
Agenda

- Agile Software Development with MSF
- Team Model
- Personas/Scenarios
- The “Agile” Pattern
- Architecture
- Testing
Roles <-> Advocacies

- Project Manager
- Architect
- Business Analyst
- Product Management
- User Experience
- Release Operations
- Development
- Test
- Release Manager
- Team of Peers
Getting the Whole Team Involved

★ Larry Sykes
Business Analyst

★ Jacqui Ackerman
Project Manager

★ Mort Gaines
Developer

★ Renee Davis
Tester

★ Art Benson
Architect

★ Ian Manning
Release Manager
Work Streams – cont.

- Capture Product Vision
- Create a Scenario
- Create a Quality of Service Requirement
- Plan an Iteration Program Management
- Create Solution Architecture
- Implement a Development Task
- Build a Product
- Test a Scenario
- Test a Quality of Service Requirement
- Fix a Bug
- Close a Bug
- Release a Product
- Guide Project
Capture Project Vision

- Write Vision Statement
- Define Personas
- Refine Personas

Business Analyst

- Brainstorm Scenarios
- Develop Lifestyle Snapshot
- Prioritize Scenario List
- Write Scenario Description
- Storyboard a Scenario

- Review Objectives
- Guide Project
- Plan an iteration

- Rent Video
- Return Video

- Brainstorm Quality of Service Requirements
- Develop Lifestyle Snapshot
- Prioritize Quality of Service Requirements
- Write Quality of Service Requirement

Create a Quality of Service Requirement

- Quality of Service Requirement List

- Rent Video
- Return Video

- Vision Statement
- Lifestyle Snapshot
- Scenario List

- Scenario
- Scenario Description
- Reports

- Quality of Service Requirement

Storyboard

Customer
Principles & Mindsets

Principles (Team)

Mindsets (Person)
Agenda

- Agile Software Development with MSF
- Team Model
- Personas/Scenarios
- The “Agile” Pattern
- Architecture
- Testing
What are Personas?

- In the context of product development, personas are descriptions of a group of typical users.

- Instead of talking about the group of users in an abstract, impersonal way, a persona represents a 'proxy' for the user group, and provides a means to talk and reason about the group through the characteristics of one fictional individual, the persona.
Where do Personas Fit?

On-site Customer

Persona

Actor
Customer Involvement

Personas → Scenarios → Working Software

Refine → Scenarios → Review

Real Customers
Example Persona - Eric

- **Role:** Online Shopper
- **Motivation:** Get it Quick
- **Usage:** Eric hates to shop but wants his equipment immediately. He will place his order on Thursday night for his weekend activity. Eric doesn’t want to wade through a catalog. Instead, he wants things that he typically orders to show immediately.
Example Persona - Anna

- **Role**: Online Shopper
- **Motivation**: Get it Cheap
- **Usage**: Anna shops for the best bargain. She looks for the best deal on similar items. She will visit half a dozen sites to find the best deal.
Agenda

- Agile Software Development with MSF
- Team Model
- Personas/Scenarios
- The “Agile” Pattern
- Architecture
- Testing
Incremental approach – Why scenarios over use cases?

- Risk reduction
  - Changes
  - Delivery
  - Progress

- UseCase:Scenario notation

RentVideo:MaxLimit
The Agile Pattern (The Engine)

Scenario List
- Scenario 1
- Scenario 2
- Scenario 3
- Scenario 4

Iteration Plan
- Iteration 1
  - Scenario 1
  - Scenario 2
  - Scenario 3
The Life of a Scenario

- Schedules Scenario
- Writes Scenario
- Divides Scenario
- Integrates Scenario
- Prioritizes Scenario
- Defines Scenario
- Tests Scenario
Remaining Work: Scenarios
Report generated: 11/04/2004 11:25 AM by someone@example.com

How much work is left and when will it be done?
View report documentation
Agenda

- Agile Software Development with MSF
- Team Model
- Personas/Scenarios
- The “Agile” Pattern
- Architecture
- Testing
The MSF Agile Architect

If you check the VSTS Guidance for an architect you’ll find:

One Work Stream “Create Solution Architecture”

- Application Diagram
- System Diagram
- Logical Datacenter Diagram
- Prototype
Shadow Architecture

- Code in Visual Studio can cast shadows

[Diagram showing class hierarchy and state validators]
Trailing Shadows

- Trailing Shadows are visual reflections of the code base

- These should come for free or nearly free
Leading Shadows

- The Architectural goal for the iteration
- Should become working code within the iteration – no BDUF
Agenda

- Agile Software Development with MSF
- Team Model
- Personas/Scenarios
- The “Agile” Pattern
- Architecture
- Testing
Getting the Whole Team Involved

Larry Sykes
Business Analyst

Jacqui Ackerman
Project Manager

Mort Gaines
Developer

Renee Davis
Tester

Art Benson
Architect

Ian Manning
Release Manager
Context Driven Testing

- Testing that is acceptable on one project may be **criminal** on another.

- Make a test plan for the project and for each iteration.
Test Thresholds

Code Coverage for Unit Test

- Which statements of my application are being executed?
- Useful to identify incomplete testing

Problems with Code Coverage

- 100% Code coverage says… NOTHING!
- 20% Code Coverage says… Something…
- Do not make it a science!
Conduct Exploratory Testing

- Become the persona...
- Make a run through the system as the persona
- Add any new bugs, scenarios or quality of service requirements discovered using this process
What happened after the show?

- Larry Sykes
  Business Analyst

- Jacqui Ackerman
  Project Manager

- Art Benson
  Architect

- Mort Gaines
  Developer

- Renee Davis
  Tester

- Ian Manning
  Release Manager
Famous last words…

- Agile solves all our problems
- If we make them do it Agile, it’ll work
- Let’s do it all at once!
Feedback

http://www.microsoft.com/msf