MathWorks
AUTOMOTIVE CONFERENCE 2019

Agile 개발 방법과 모델기반 설계

이영준
Agenda

- Agile Values and Typical Workflow
- Model-Based Design (MBD)
- Agile Development with MBD
- Scrum with MBD
### Agile Values

<table>
<thead>
<tr>
<th>Values</th>
<th>Over</th>
<th>Principles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Individuals &amp; Interactions</td>
<td>Process and Tools</td>
<td></td>
</tr>
<tr>
<td>Customer Collaboration</td>
<td>Contract Negotiation</td>
<td></td>
</tr>
<tr>
<td>Working Software</td>
<td>Comprehensive Documentation</td>
<td></td>
</tr>
<tr>
<td>Responding to Change</td>
<td>Following a Plan</td>
<td></td>
</tr>
</tbody>
</table>

“While there is value in the items on the right, we value the items on the left more.”

- The Agile Manifesto, 2001
Agile: Values, Principles and Practices

Agile is a mindset defined by values, guided by principles and manifested through many different practices. Agile practitioners select practices based on their needs.

~ Agile Practice Guide (PMI® and Agile Alliance®)
Typical agile development workflow
Model-Based Design
Models $==$ Understanding
Simulation

Physical Prototyping
Simulation is key to Level 4-5 autonomy

*Source: Center for Artificial Intelligence, Saarland University
Model-Based Design

Systematic use of models throughout the development process

Modeling
Simulation

Fast repeatable tests
Systematic use of models throughout the development process
Model-Based Design

Systematic use of models throughout the development process

- Modeling
  - Simulation
  - Fast repeatable tests
- Automation
  - Coding
  - Verification
Model-Based Design

Systematic use of models throughout the development process

Modeling
- Simulation

Automation
- Coding
- Verification

Fast repeatable tests
Fast agile development loops
Model-Based Design

Executable Specification
- Unambiguous – easy to understand
- Systems engineering - modeling whole system including environment
- Sharing of models to improve communication and collaboration
- Early validation and test development

Multi-domain Design
- Model algorithms and environment
- Perform integration testing at model level before implementation

Automatic Code Generation
- Eliminate errors from hand-coding
- Regenerate easily for different targets

Continuous Test and Verification
- Detect errors early in development
- Reduce use of physical prototypes
- Reuse tests throughout development process
Agile Development with MBD
Agile Development – Key Principles

Inputs from Executives, Team, Stakeholders, Customers, Users

Product Owner

The Team

Sprint Backlog

Product Planning Meeting

Team selects starting at top as much as it can commit to deliver by end of Sprint

1-4 Week Sprint

Sprint Backlog

Sprint Planning Meeting

Sprint Review

Finished Work

Sprint Retrospective

Sprint end date and team deliverable do not change

Completed

Burndown/up Charts

Every 24 Hours Daily Scrum Meeting

Customer satisfaction by rapid delivery of useful software

Picture Source: http://www.softwareplant.com/scrum/
Principle: Customer satisfaction by rapid delivery of useful software

- **Simulation** allows customer evaluation of functional behavior **early and often**.

- **Useful software** can be delivered throughout the project via **code generation**.

![Diagram of Gamma Correction](image)
Agile Development – Key Principles

Customer satisfaction by rapid delivery of useful software

Welcome changing requirements, even late in development

Picture Source: http://www.softwareplant.com/scrum/
Principle: Welcome changing requirements, even late in development

- **Requirements traceability** supports **impact analysis** of affected model components.

- Dynamic/executable models allow for **rapid evaluation of requirements** changes.

- **Regression testing** of simulation-based tests can be automated to confirm **new design meets existing requirements**.
Agile Development – Key Principles

Customer satisfaction by rapid delivery of useful software

Welcome changing requirements, even late in development

Working software is the principal measure of progress
Principle: Working software is the principal measure of progress

- Functional designs can be evaluated continuously via **executable models**.

- Code generation supports rapid software deliveries, **rapid prototyping** and **HIL test**
Agile Development – Key Principles

Customer satisfaction by rapid delivery of useful software

Continuous attention to technical excellence and good design

Welcome changing requirements, even late in development

Working software is the principal measure of progress

Picture Source: http://www.softwareplant.com/scrum/
Principle: Continuous attention to technical excellence and good design

- Continuous development and testing in an executable modeling environment allows for – “build a little / test a little” workflows.

Where are you! ...still coding?

Work on your functionality, not on your code

Model and Simulate Your System  Test Early and Often  Automatically Generate Code
Agile Development – Key Principles

- Customer satisfaction by rapid delivery of useful software
- Welcome changing requirements, even late in development
- Continuous attention to technical excellence and good design
- 1-4 Week Sprint
- Simplicity
- Burndown/up Charts
- Every 24 Hours
- Daily Scrum Meeting
- Sprint Backlog
- Task Breakout
- Sprint Planning Meeting
- Sprint end date and team deliverable do not change
- Working software is the principal measure of progress

Picture Source: http://www.softwareplant.com/scrum/
Principle: Simplicity—the art of maximizing the amount of work not done—is essential

- Model-Based Design supports identification of bugs where they are introduced and “cheaper” to fix.

- Reduces rework (design and test).
Agile Development – Key Principles

- **Customer satisfaction by rapid delivery of useful software**
- **Welcome changing requirements**, even late in development
- **Self-organizing teams**
- **Continuous attention** to technical excellence and good design
- **Working software** is the principal measure of progress
- **Simplicity**
- **Burndown/up Charts**
- **Every 24 Hours**
- **Daily Scrum Meeting**
- **Sprint Review**
- **Finished Work**
- **Sprint Retrospective**
- **At regular intervals, the team reflects on how to become more effective**
- **Inputs from Executives, Team, Stakeholders, Customers, Users**
- **Product Owner**
- **Product Backlog**
- **Ranked list of what is required: features, stories, ...**
- **Sprint Planning Meeting**
- **Team selects starting at top as much as it can commit to deliver by end of Sprint**
- **Sprint Backlog**
- **Sprint end date and team deliverable do not change**
- **Task Breakout**

Picture Source: http://www.softwareplant.com/scrum/
Principle: At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly

- Collect metrics from models to measure and improve process e.g. test coverage, model metrics
- Flexible tool suite supports wide variety of workflows and development processes.
- Open APIs.
Agile Development – Key Principles

- **Customer satisfaction by rapid delivery of useful software**
- **Welcome changing requirements, even late in development**
- **Continuous attention to technical excellence and good design**
- **Self-organizing teams**
- **Working software is the principal measure of progress**
- **Simplicity**
- **At regular intervals, the team reflects on how to become more effective**
- **Close, daily cooperation between business people and developers**
- **Customer satisfaction by rapid delivery of useful software**

*Inputs from Executives, Team, Stakeholders, Customers, Users*
Scrum with MBD
Model-Based Design
- Executable Specifications from Models
- Design with Simulation
- Implementation with Automatic Code Generation
- Continuous Test and Verification
MIL – Modeling only

Model-Based Design
- Executable Specifications from Models
- Design with Simulation
- Implementation with Automatic Code Generation
- Continuous Test and Verification

Evolved System Requirements
- Integrate subsystems
- Verify design choices

Develop/Design plant and algorithm models
- Integrate external code
- Test
- Generate code
- Evaluate/Decide
- Validation
HIL Verification

Model-Based Design
- Executable Specifications from Models
- Design with Simulation
- Implementation with Automatic Code Generation
- Continuous Test and Verification

Integrate external code

Integrate candidate with HIL system

Generate code of plant model

Verify candidate implementation against simulated plant

Develop/Design plant model

Evolved System Requirements

BACKLOG
DEFINITION OF READY
DEFINITION OF DONE
DELIVERY

Simulate
Test
Rapid prototyping

Integrate with real system

Verify prototype against real plant

Evolved System Requirements

Automatic code generation
Real-Virtual simulation

Generate code of algorithm to prototype board

Integrate external code

Develop/Design algorithm

Simulate  Test

BACKLOG

DEFINITION OF READY

DEFINITION OF DONE

DELIVERY

Model-Based Design

- Executable Specifications from Models
- Design with Simulation
- Implementation with Automatic Code Generation
- Continuous Test and Verification

Generate

code
of
algorithm
to
prototype
board

Integrate
external
code

Evolved System Requirements

Develop/Design
algorithm

Simulate  Test
Integrated Agile System Development through Rapid Development Iterations

- **Deploy**
  - Desktop CPU: MIL, SIL
- **Design**
  - Target CPU: PIL
  - Prototyping RT CPU: RCP, HIL
  - Production EC: HIL, OTRP

- **Integrate**
  - Algorithm
  - Plant

- **Implement**
  - External Code
    - Models (Components, Subsystems, System)

<table>
<thead>
<tr>
<th>Controller</th>
<th>Plant</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIL, SIL, PIL</td>
<td>Virtual, Virtual</td>
</tr>
<tr>
<td>RCP</td>
<td>Virtual RT, Real</td>
</tr>
<tr>
<td>OTRP</td>
<td>Real, Real</td>
</tr>
<tr>
<td>HIL</td>
<td>Real, Virtual RT</td>
</tr>
</tbody>
</table>

**Integrated Agile System Development through Rapid Development Iterations**
Who will be successful in the future?

**Mechanical-centric**

**Model-centric**

**Software-centric**

Comprehensive models
Simulation based testing
Generate code and automate verification
Questions?