Lean Principles

presented by
Werner Wild & Barbara Weber

IT Cluster Tirol – 4.6.2009

Contents

- Origins of Lean Development
  - The Toyota Production System
- Principles of Lean Thinking
  - **Eliminate Waste**
  - **Create Knowledge**
  - **Defer Commitment**
  - **Deliver Fast**
  - Built Quality In
  - Respect People
  - Optimize the Whole

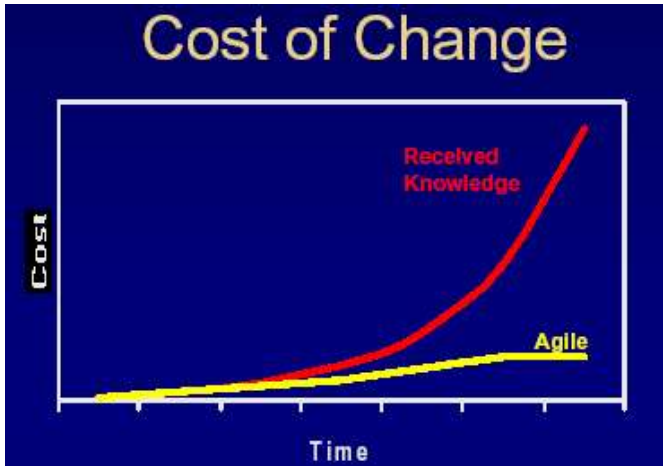
The Toyota Production System

Taiichi Ohno

1912-1990

- Just-in-time flow
  - Non stock production
  - Build only what is needed
  - Eliminate everything which does not add value
- Automation
  - Stop if something goes wrong
  - Zero Inspection

Changing the Mental Model



- Received Knowledge:
  - Die Change is Expensive
  - Don't Change Dies
- Taiichi Ohno
  - Economics Requires Many Dies Per Stamping Machine
  - *One Minute Die Change*

- Received Knowledge:
  - Code Change is Expensive
  - Freeze Design Before Code
- The Agile Imperative
  - Economics Requires Frequent Change In Evolving Domains
  - *Last Responsible Moment*

Principles of Lean Thinking

1. **Eliminate Waste**
2. Create Knowledge
3. Defer Commitment
4. Deliver Fast
5. Build Quality In
6. Respect People
7. Optimize the Whole

Principle 1: Eliminate Waste

- Taiichi Ohno on how the Toyota Production System works:

  All we are doing is looking at the timeline from the moment a customer gives us an order to the point when we collect the cash. And we are reducing this timeline by removing the non value-adding wastes.

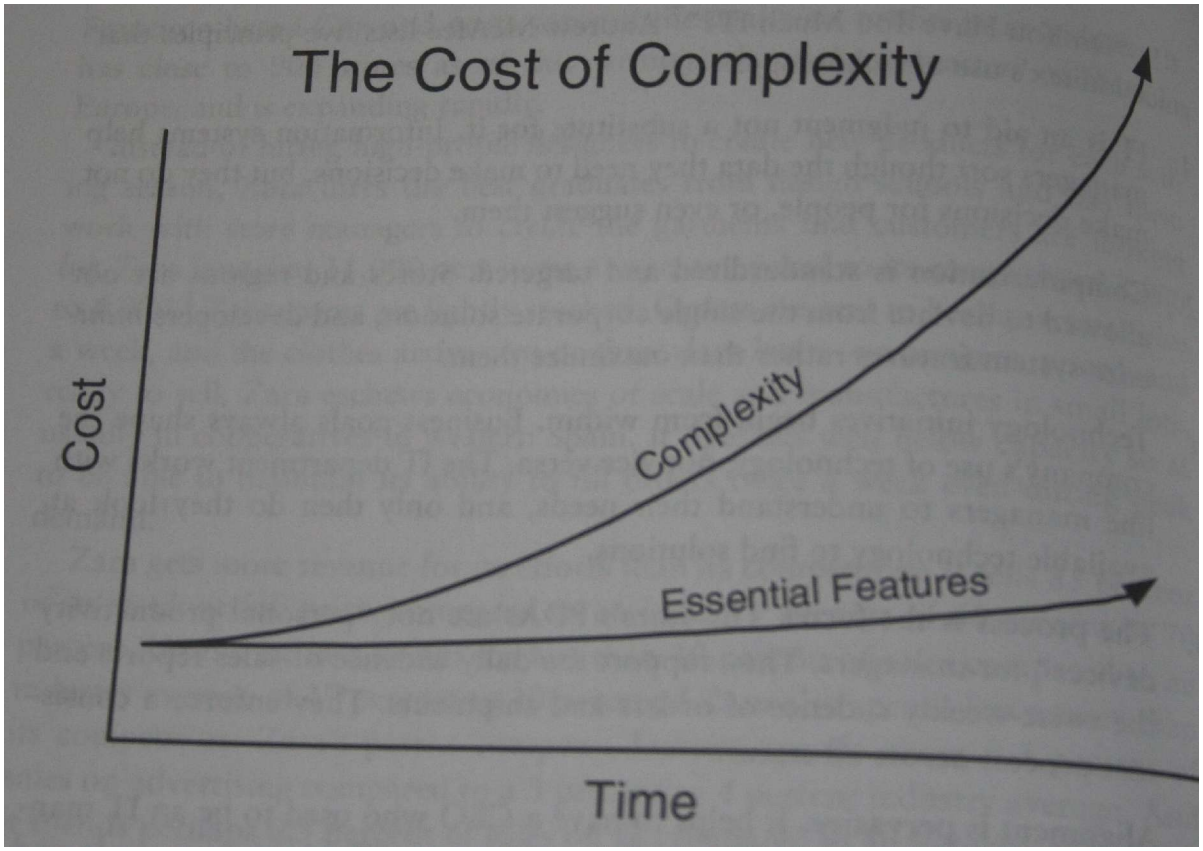
Eliminate Waste

- Waste
  - Anything that does not create value for the customer
  - The customer would be equally happy with the software without it

- Prime Directive of Lean Thinking
  - Create ***Value*** for the customer
  - Improve the ***Value Stream*** by removing non value-adding activities

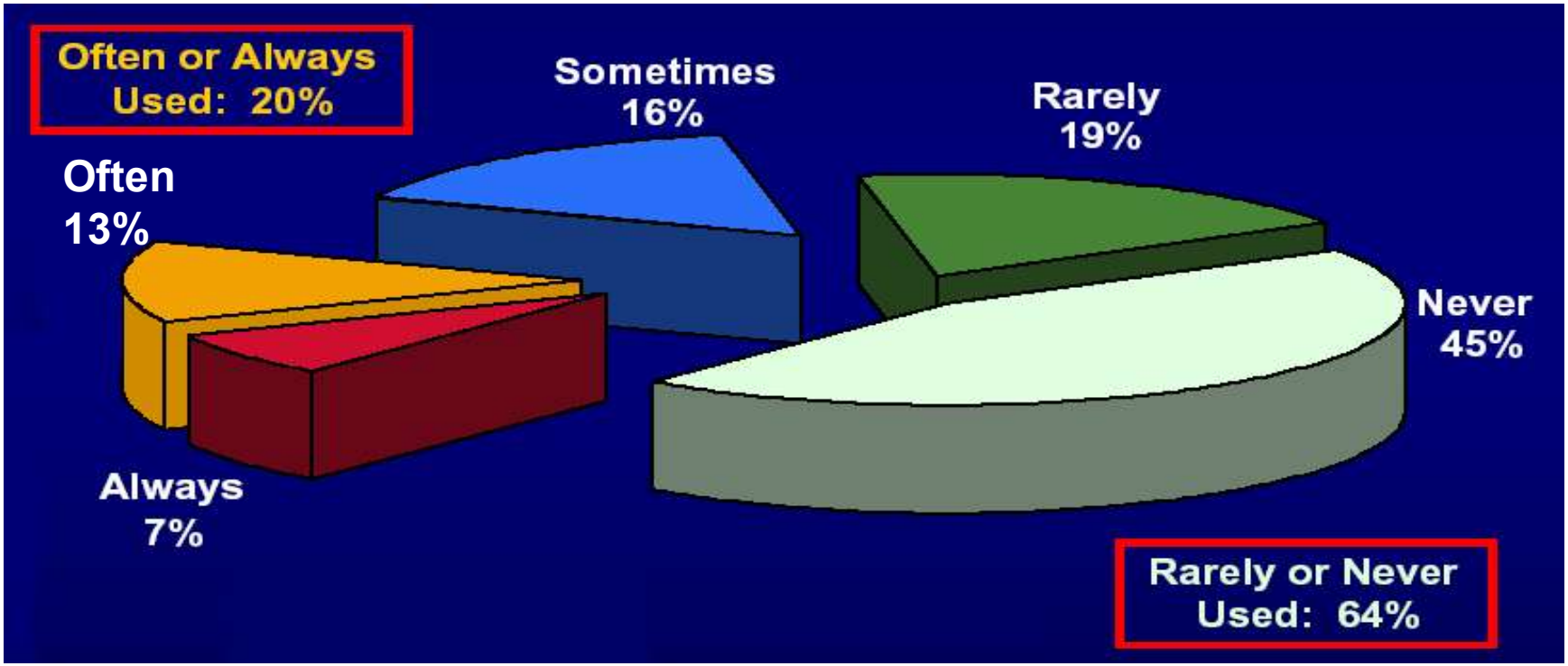
Complexity

- Complexity is the biggest source of waste



The biggest Source of Waste

Features and Functions Used in a Typical System

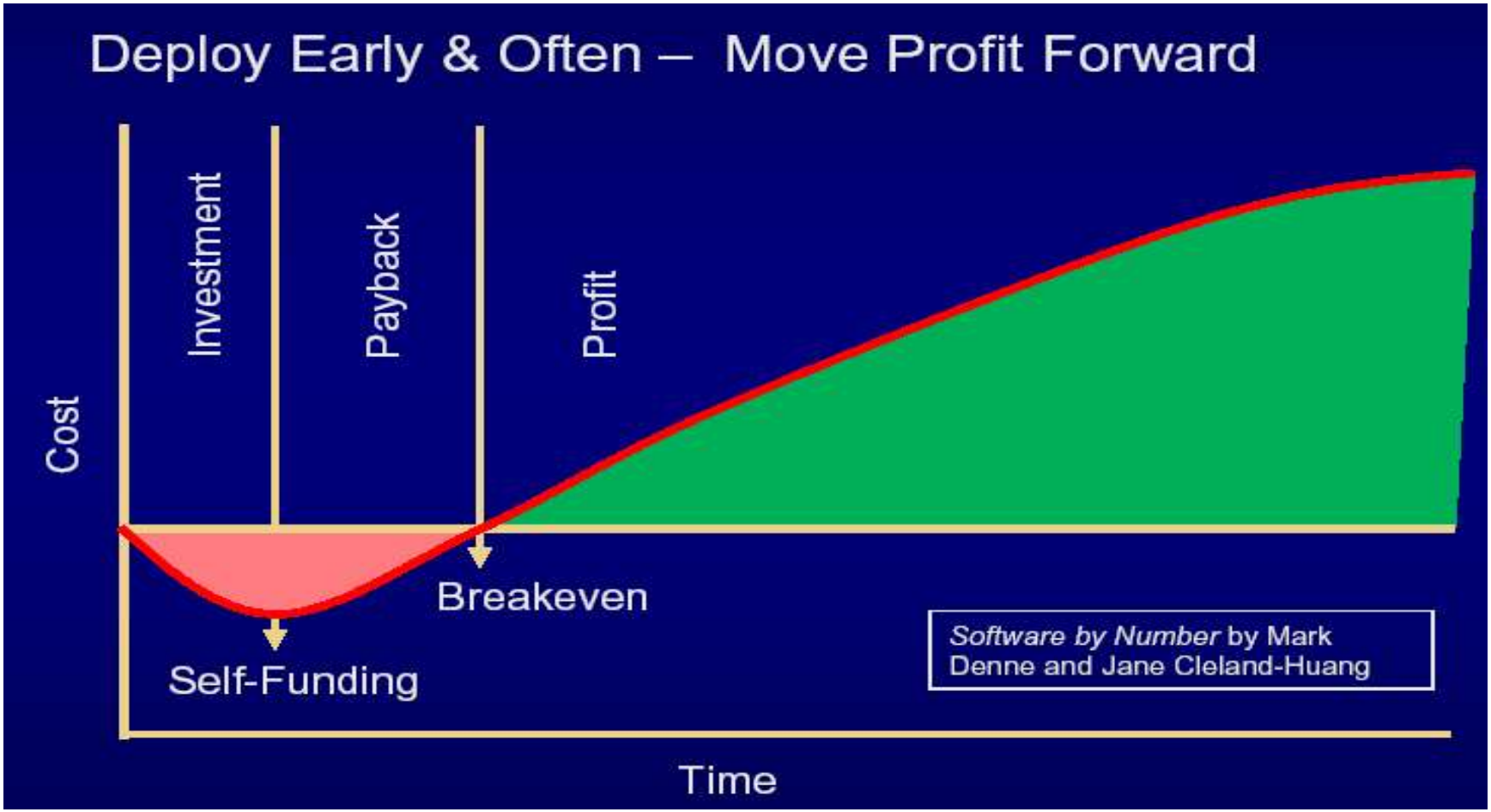


Standish Group Study Reported at XP2002 by Jim Johnson, Chairman

Write Less Code

- Justify every feature
  - Limit the features and the functions that make it into the code base
  - Every feature should proof that if will create more economic value than its lifecycle costs
- Minimum useful feature sets
  - Divide software into minimum useful features and deploy these one set at a time, highest priority first
  - Allows to use the software much faster

Minimum Useful Feature Sets



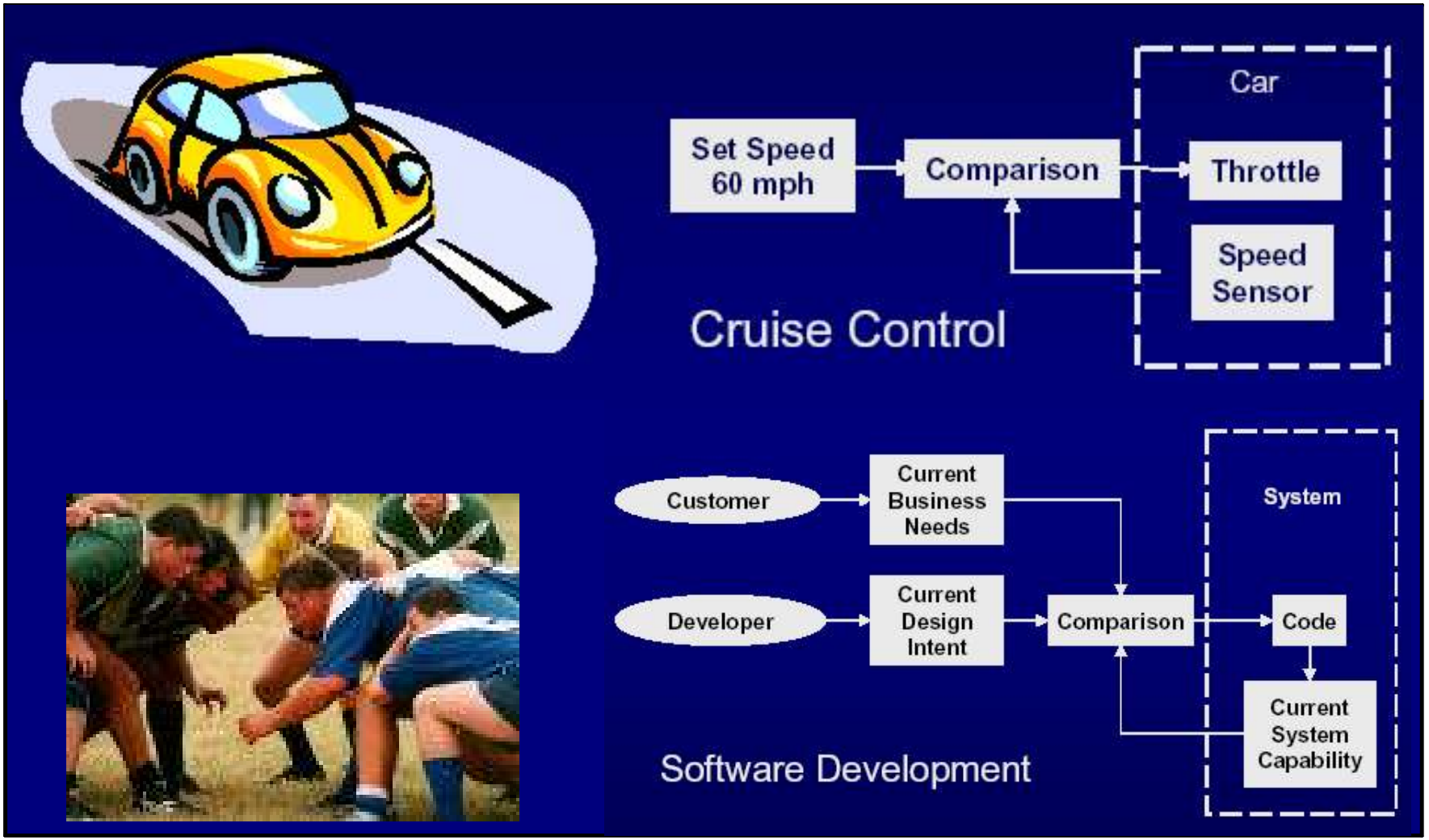
1st step is Seeing Waste

Manufacturing	Software Development
In-Process Inventory	Partially Done Work
Over-Production	Extra Features
Extra Processing	Relearning
Transportation	Handoffs
Motion	Task Switching
Waiting	Delays
Defects	Defects

Principles of Lean Thinking

1. Eliminate Waste
2. **Create Knowledge**
3. Defer Commitment
4. Deliver Fast
5. Build Quality In
6. Respect People
7. Optimize the Whole

Principle 2: Create Knowledge



Taking Feedback into Account

- Waterfall approach is based on the idea that knowledge in the form of “requirements” exists prior to and separate from coding
- Software development is a knowledge-creation process; successful software products emerge
- Early design does not take ongoing feedback that comes from actually building software into account

Principles of Lean Thinking

1. Eliminate Waste
2. Create Knowledge
3. **Defer Commitment**
4. Deliver Fast
5. Build Quality In
6. Respect People
7. Optimize the Whole

Defer Commitment

- The longer we defer decisions, the more we can learn
- Challenge to make decisions Just-in-time
- 2 Approaches for learning
  - Building a system which is change tolerant
    - Refactoring
  - Building several options
    - Set-Based Design

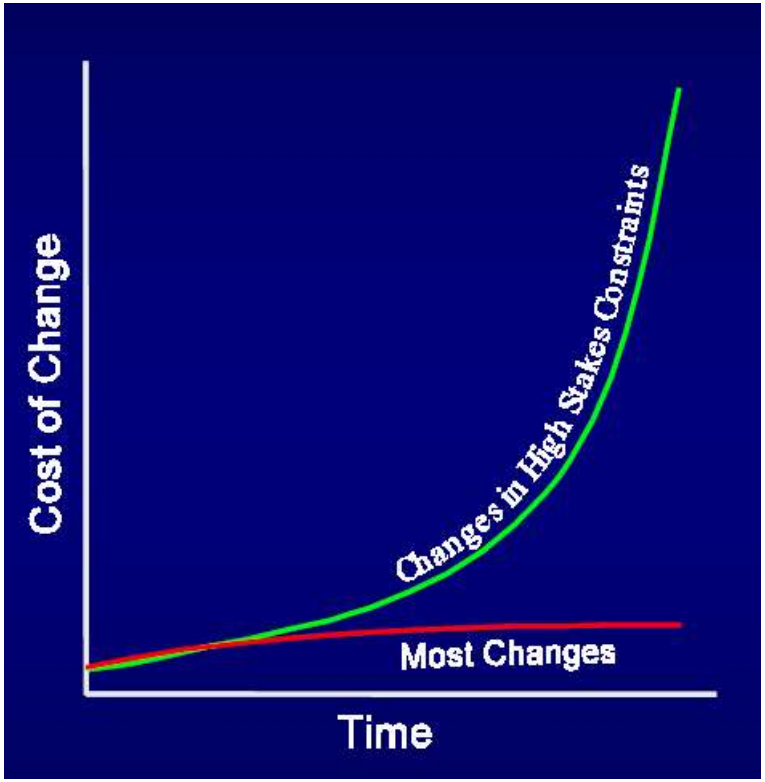
Set-Based Design

- Is based upon the idea of building several options
- Leader with technical expertise knows where to maintain options
- At the **last responsible moment** the option that gives the best overall solution is chosen
- Appropriate approach for making high-impact, irreversible decisions
- Example: Prius Engine System

Defer Commitment

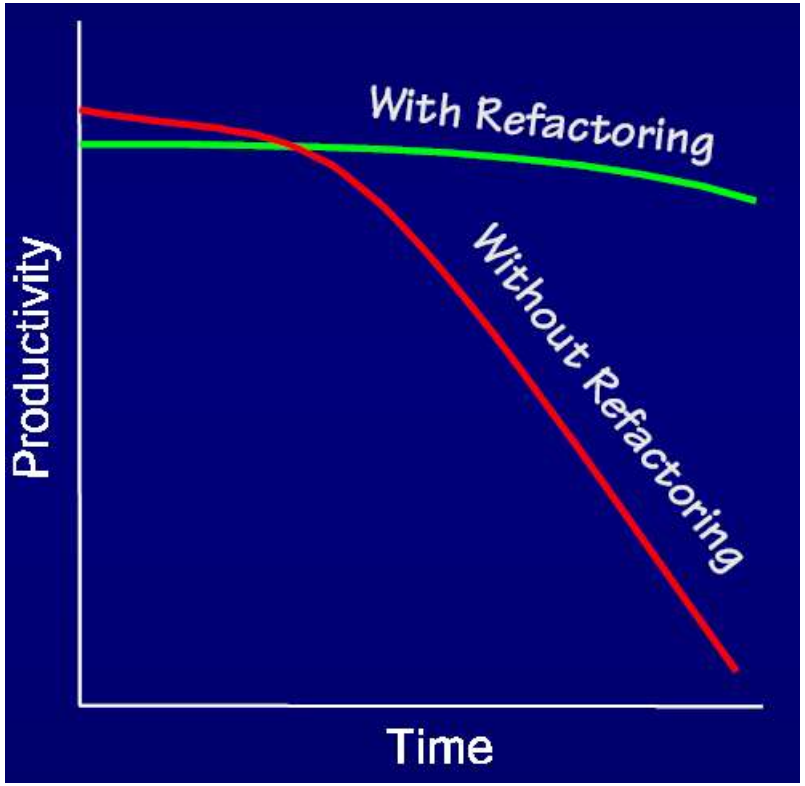
Two Kinds of Change

- High Stakes Constraints
  - Examples:
    - Language
    - Layering
    - Usability
    - Security
    - Scalability
  - Rule:
    - Only a Few
    - At a High Level
- Most Changes
  - Keep the Cost Low!



Refactoring

- Building a change tolerant code base which allows to adapt to changes
- Mitigates the risk and minimizes the cost of complexity in the code base
- Requires automated testing, continuous integration and stop-the-line



Principles of Lean Thinking

1. Eliminate Waste
2. Create Knowledge
3. Defer Commitment
4. **Deliver Fast**
5. Build Quality In
6. Respect People
7. Optimize the Whole

Principle 4: Deliver Fast

- The most disciplined organizations are those that respond to customer requests
  - Rapidly
  - Reliably
  - Repeatedly

- Software Development Maturity
  - The speed at which you reliably and repeatedly convert customer requests to deployed software
  - Quality of a software development process can be measured by measuring the average end-to-end cycle time of the development process

Queues

- Cycle Time
  - Average End-to-End Process Time
    - From Entering The Terminal
    - To Arriving at the Gate
- Time Spent in a Queue is Wasted Time
- The Goal: <u>Reduce Cycle Time</u>

Queuing Theory

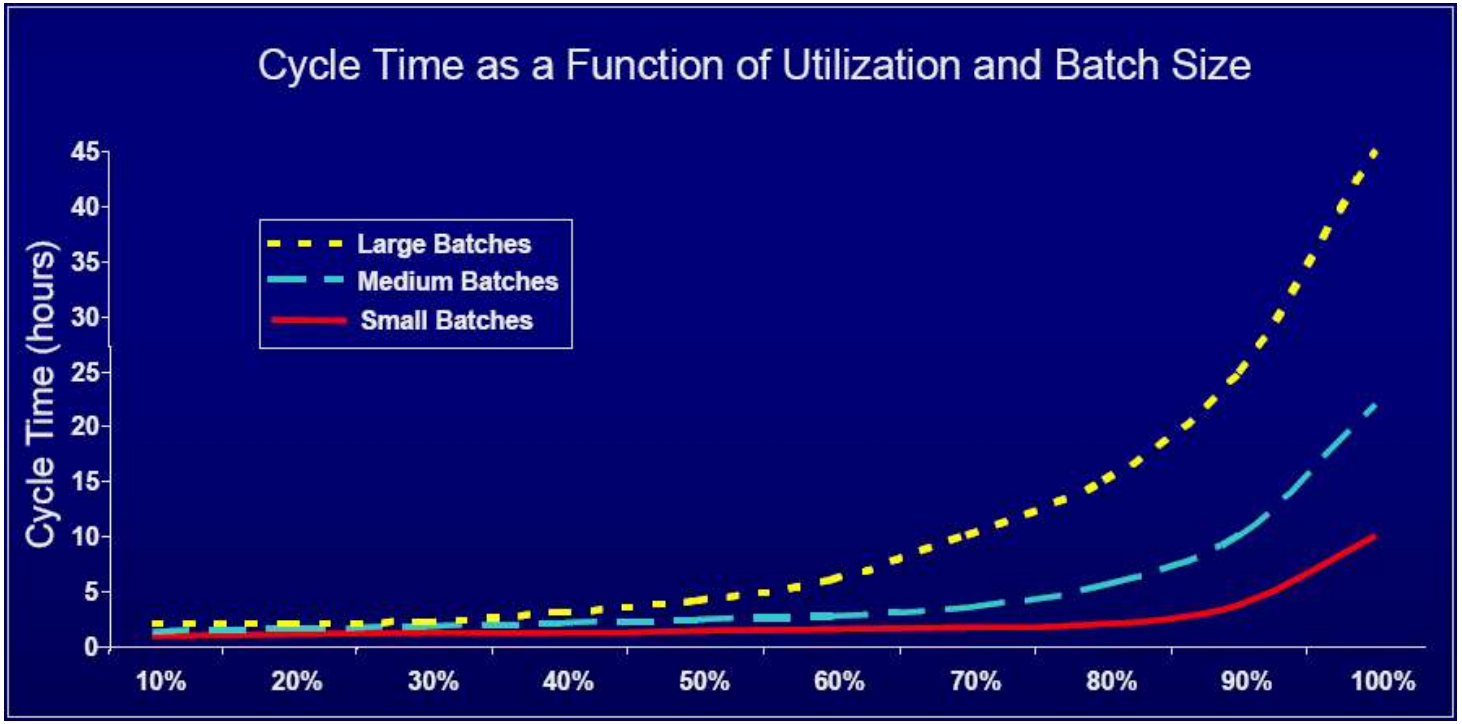
- Little's Law

$$\text{Cycle Time} = \frac{\text{Things in Process}}{\text{Average Completion Rate}}$$

- 2 ways to reduce cycle time
  - Do things faster
  - Reduce the number of things in the process

Queuing Theory Lessons

- Small Batches Move Faster
- Slack Resources Decrease Cycle Time



References

- Mary and Tom Poppendieck: An Agile Toolkit for Software Development Managers. Addison Wesley, 2003.

- Mary and Tom Poppendieck: Implementing Lean Software Development. From Concept to Cash. Addison Wesley, 2006.

Acknowledgements

Some of the slides are based on the Tutorial “Lean Software Development” by Mary & Tom Poppendieck and were modified by Barbara Weber.

I would like to thank Mary & Tom & Barbara for their kind permission to use their slides in this presentation.

Thanks for your Attention !

Contacts

werner.wild@evolution.at

barbara.weber@uibk.ac.at