

# CROSSTALK

April 2007

The Journal of Defense Software Engineering

Vol. 20 No. 4

## *7 Ways to Fail with Scrum!*



**Deep Agile**  
SOFTWARE DEVELOPMENT

Jeff Sutherland, Ph.D.  
Co-Creator of Scrum

<http://jeffsutherland.com/scrum>

# AGILE DEVELOPMENT

# Plan driven development

- High failure rate
- Produces software that sucks
  - Fails to fit customer needs
  - High defect rate
- Over 50% waste
- Delays time to market
- Poor working environment

# Value driven development

- High success rates
- Produces software that meets customer needs
- Minimal waste
- Accelerates early revenue
- Improves working environment

# Fail by not doing Agile Development

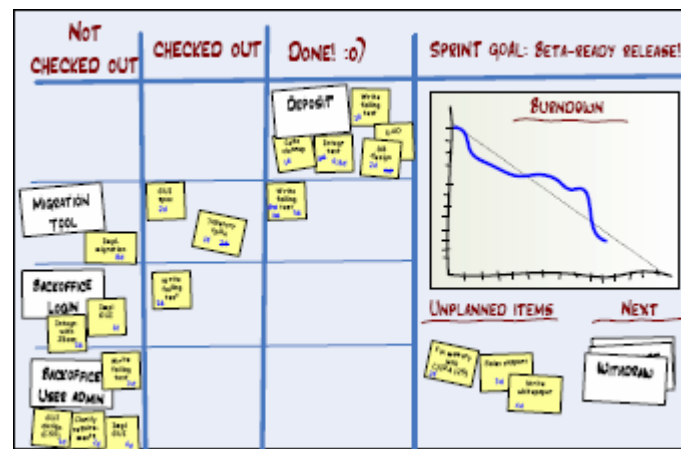
- QCON 2007 - 135 people polled
  - 30 said they were doing Scrum
- 50% were not doing Agile development
  - Failure to meet Nokia test for iterative development
    - Iterations longer than 6 weeks
    - Iterations not timeboxed
    - Team tried to finish all specification before programming
    - Iterations do not result in workable code
    - Iterations do not include testing

# For those not doing Scrum failure is expected

- QCON 2007 - 30 said they were doing Scrum
- 50% were not doing iterative development
- Only 3 out of 30 met the Nokia test for Scrum
  - The team knows who the product owner is
  - The product backlog exists and contains estimates
  - The team can generate a release burn-down chart and knows their velocity
  - There are no project managers in the project disrupting the work of the team

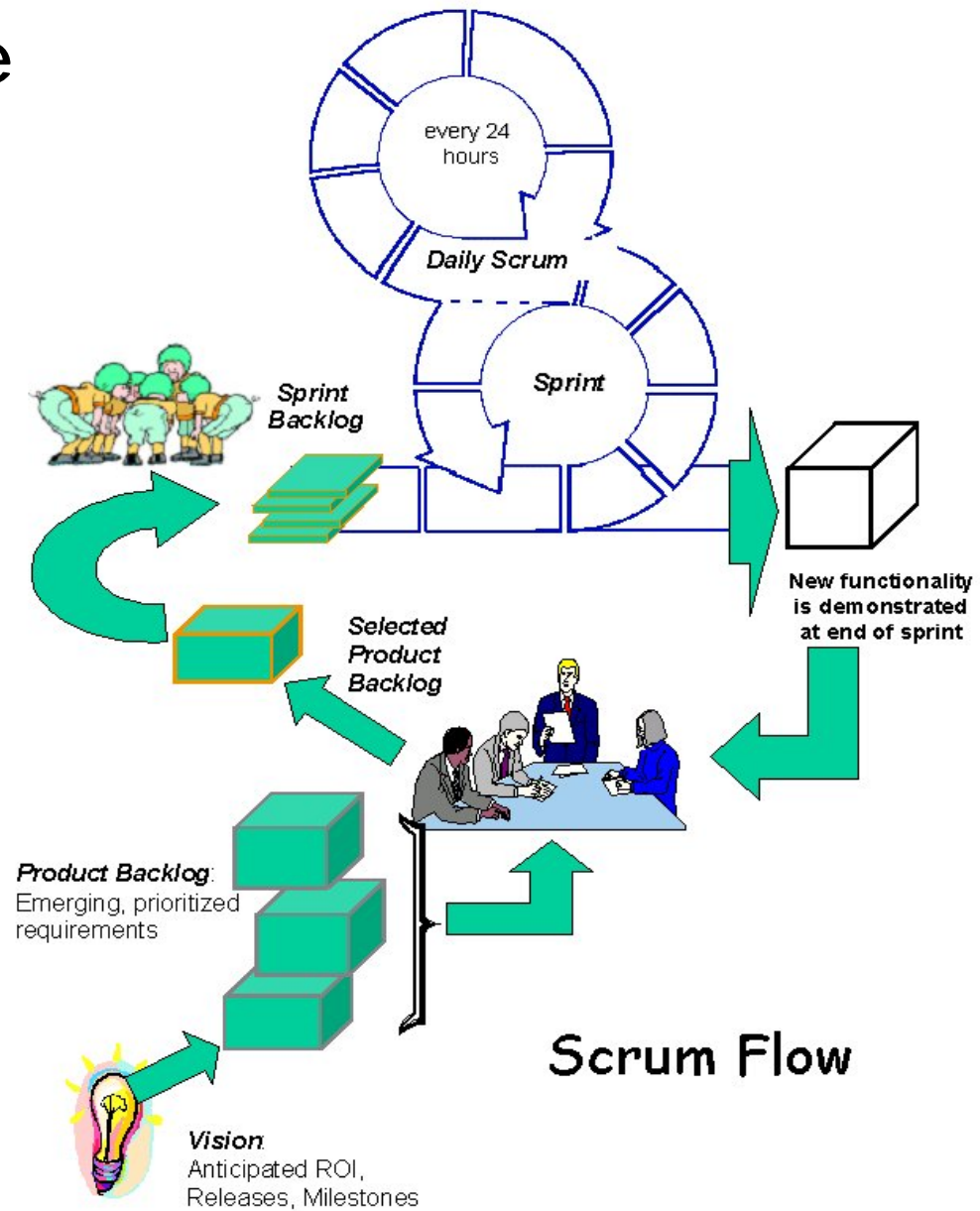
# For those of you doing Scrum

- You know who the product owner is
- There is a product backlog prioritized by business value
- The product backlog is has estimates created by the team
- The team generates burndown charts and knows their velocity
- There are no project managers (or anyone else) disrupting the work of the team



*Kniberg, Henrik. Scrum and XP from the Trenches: How We Do Scrum. Version 2.1, Crisp, 5 Apr 2007.*

# Scrum Failure Points



# 1. Product Owner failure point

- The Product Owner does not have:
  - A vision
  - A business plan
  - A release roadmap
- The Product Backlog:
  - Is not ordered properly
  - Does not contain all work (including technical issues)
  - Is not ready for the Sprint planning meeting
    - Is not sized properly
    - Is not estimated properly
    - Does not have enabling specifications
- The Product Owner is AWOL during the Sprint



## 2. Sprint Planning failure point

- The Product Owner does not communicate clearly:
  - The vision, the business plan, the release roadmap
  - The Product Backlog is not ready
- The Team:
  - Takes too much off the Product Backlog
  - Does not break down features into Sprint tasks with good estimates – Sprint Backlog
- The ScrumMaster does not make sure the original estimates in the Product Backlog equals the more detailed estimates in the Sprint Backlog
- Trust, transparency, and truth is not present
- The plan does not meet the three finger test



# 3. Daily Meeting failure point

- Team is more than 7 plus or minus 2 people
- Every person does not speak
- Meaningful information is not communicated
  - Tasks started, stopped, completed
  - Estimates expanding
  - Impediments including personal issues
- Team does not self-organize
  - Must replan work based on information heard
  - Use 60 second rule to eliminate most impediments in the meeting
- ScrumMaster runs a lousy meeting
  - Longer than 15 minutes
  - Lack of dynamic facilitation

## 4. ScrumMaster failure point

- ScrumMaster not dedicated and focused on team
  - Update burndown daily
  - Remove impediments daily
  - Deal with personal issues daily
- Lack of good personal and facilitative skills
  - Communicate, communicate, communicate
  - Listen, listen, listen
- Lack of good leadership skills
  - Tolerates distrust, lying (even if by omission), and hiding information
  - Fails to have a prioritized impediment list and eliminate impediments
  - Fails to deal with personal dynamics and other personal problems

## 5. Team failure points

- Lack of required technical or domain knowledge
- Failure to produce burndown, remove impediments, and increase velocity
- Working on anything not on the Sprint backlog
- Individual multitasking
- Team generating excessive work in progress
- Failure to test early
- Failure to improve engineering practices
- Lack of focused and dedicated resources

## 6. Sprint Review failure points

- Cannot demonstrate tested, working code
- Software is not done
  - No consistent definition of done
  - Product Owner does not verify features are done
  - If not done, Product Owner does not reprioritize on the product backlog
- Velocity is not clear from Sprint results
- Team does not do retrospective
  - Team does not act on retrospective to improve
  - Too much churn in team composition at end of Sprint makes it impossible to execute on retrospective recommendations.

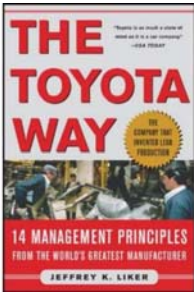
# 7. Finally – management failure!

- Failure to have a business model that works
- Failure to provide adequate resources
- Failure to smooth out flow - Mura
- Failure to avoid stressing system - Muri
  - Violating sustainable pace
  - Disrupting teams during Sprint
- Failure to eliminate waste – Muda
- Failure to eliminate any impediments the team cannot eliminate
- *Failure to challenge teams to move beyond mediocrity*

# The Bottom Line

- Avoiding these failure modes is guaranteed to:
  - Reduce planning costs by 80%
  - Reduce product delivery and deployment costs by 50%-80%
  - Improve quality by 166-1200%
  - Improve customer satisfaction dramatically
  - Enhance the working lives of development team
  - Produce better software, often best in class





# The Challenge

- In order to avoid these failure modes your company mission statement should be like Toyota's:
  - Improve the life of the community (the company, stakeholders, and the environment)
    - Google mantra – do no evil!
  - Improve the life of the employees by making the workplace creative, innovative, fun, and productive
    - Do less to produce more
  - Improve the life of the customers by delivering real value
- Only then will increased revenue, market share, and profitability follow you all the days of your life!

*Liker, J. K. (2004). The Toyota way : 14 management principles from the world's greatest manufacturer. New York, McGraw-Hill.*





Questions?

# Bibliography

- Cohn, M. (2005). Agile Estimation and Planning, Addison-Wesley.
- Larman, Craig and Basili, Vic. Iterative and Incremental Development: A Brief History. IEEE Computer, [June 2003 \(Vol. 36, No. 6\)](#) pp. 47-56
- Liker, J. K. (2004). The Toyota way : 14 management principles from the world's greatest manufacturer. New York, McGraw-Hill.
- Poppendieck, M. and T. Poppendieck (2006). Lean Software Development: An Implementation Guide, Addison-Wesley.
- Kniberg, Henrik. Scrum and XP from the Trenches: How We Do Scrum. Version 2.1, Crisp, 5 Apr 2007.
- Sutherland, J., C. Jacobson, et al. (2007). Scrum and CMMI Level 5: A Magic Potion for Code Warriors! Agile 2007, Washington, D.C., IEEE.
- Sutherland, J. and K. Schwaber (2007). The Scrum Papers: Nuts, Bolts, and Origins of an Agile Method. Boston, Scrum, Inc.
- Takeuchi, H. and I. Nonaka (1986). "The New New Product Development Game." Harvard Business Review(January-February).
- Takeuchi, H. and I. Nonaka (2004). Hitotsubashi on Knowledge Management. Singapore, John Wiley & Sons (Asia).