9th Annual Fall ASQ Conference

Risk Mitigation during Software Development

Presented by
Samuel Prasad, PhD, PMP, CSQE
Technology Management Consultant

Farmingdale State College
Gleeson Hall
2350 Broadhollow Rd.
Farmingdale, NY 11735-1021

November 4, 2011

Why is risk mitigation important?

Increase the odds of success!!

- **Identify high risk components within an application that are more likely to be error prone.**
- **Identify high value components within an application to limit financial damage.**

When does risk assessment start?

At the beginning of the project. How do you know when a project is "high-risk"?

- **Project size is directly proportional to risk. The larger the project in terms of cost, time, scope and number of functional areas involved, the greater the risk.**
- **Technology experience is inversely proportional to project risk. The more experience the development team has, less is the risk.**

Characteristics of a high-risk project

- Specifications are not fully defined.
- Project affects more than one business area or crosses organizational boundaries.
- Project cannot be completed in less than 6 months.
- Team is unfamiliar with chosen methodology, technology and/or architecture.
- Business commitment is low.

Specifications are not adequately defined

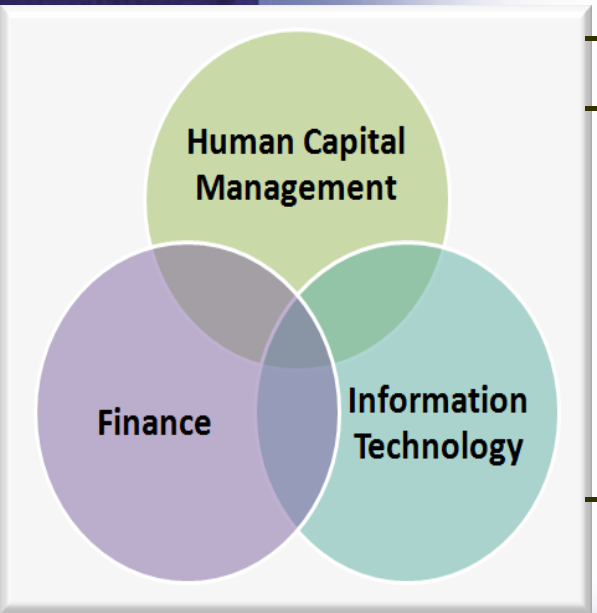
- **Reason:**
  - **Users are not sure what they want.**

- **Solution**
  - **Lean-Agile (Scrum)**
  - **Create wireframes to demonstrate navigation and capabilities.**
  - **Identify capabilities that have the highest value to the user.**
  - **Time-box development and pick the list of product capabilities based on value that can be implemented in the allotted time.**

Affects more than one business area

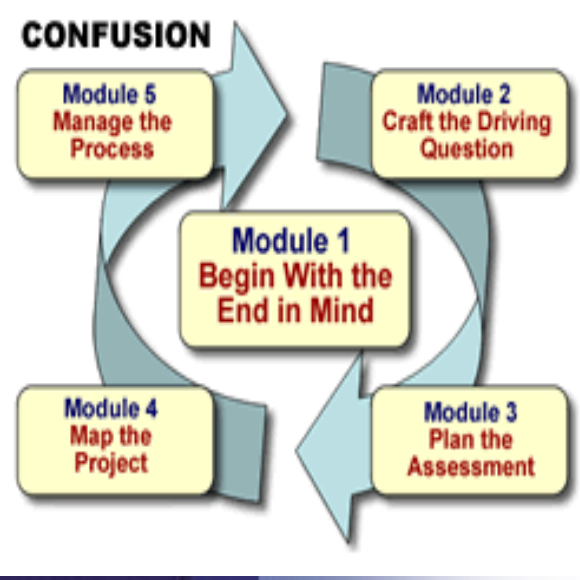
- **Reason**
  - **Inter-dependencies are not fully known.**
  - **Effects on processes and personnel in other areas have not been studied.**

- **Solution**
  - **Lean-Agile (Scrum)**
  - **Form cross functional team.**
    - **Business sponsor – user/customer.**
    - **Product owner(s) from all areas.**
    - **Scrum master.**
    - **Engineering team with required skills in the "performing" phase.**
  - **Scrum master runs JAD sessions.**



Project cannot be completed in less than 6 months.

- **Reason**
  - **Product roadmap has not been clearly flushed out.**



Solution

- **Lean-Agile (Scrum)**
- **Define product by capabilities, not based on tasks.**
- **Identify capabilities that are of most value to the user.**
- **Each capability must be delivered to the user within 2 to 4 weeks based on team velocity.**

Developers are unfamiliar with the chosen technology

- **Reason**
  - Use of bleeding-edge, unproven technology.
  - Engineers are unfamiliar and not trained in the new technology.
- **Solution**
  - Hire an experienced consultant to lead the project.
  - Turn the project over to a vendor who specializes in the technology.
  - Train project team in the chosen technology.
  - Design tests to specifically assess the effectiveness and suitability of the new technology.

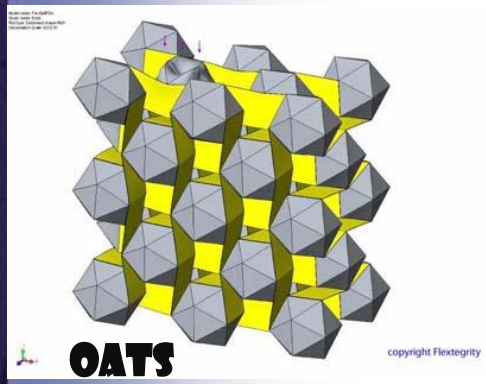
Business commitment is low

- **Reason**
  - **Product value to the business is questionable.**
- **Solution**
  - **Get product sponsors to champion the project.**
  - **Otherwise kill the project. Get out!!**

Summary

- Learned to identify some of the signs that make a project inherently high-risk.
- Defined probable reasons for the high-risk nature of a project.

How do you create working software?



WORKING SOFTWARE

SCRUM

TDD

ALL CODE IS GUILTY UNTIL PROVEN INNOCENT

CODESMACK

Lean-Agile (Scrum)

- Team fully committed

- Focus on working software at every iteration

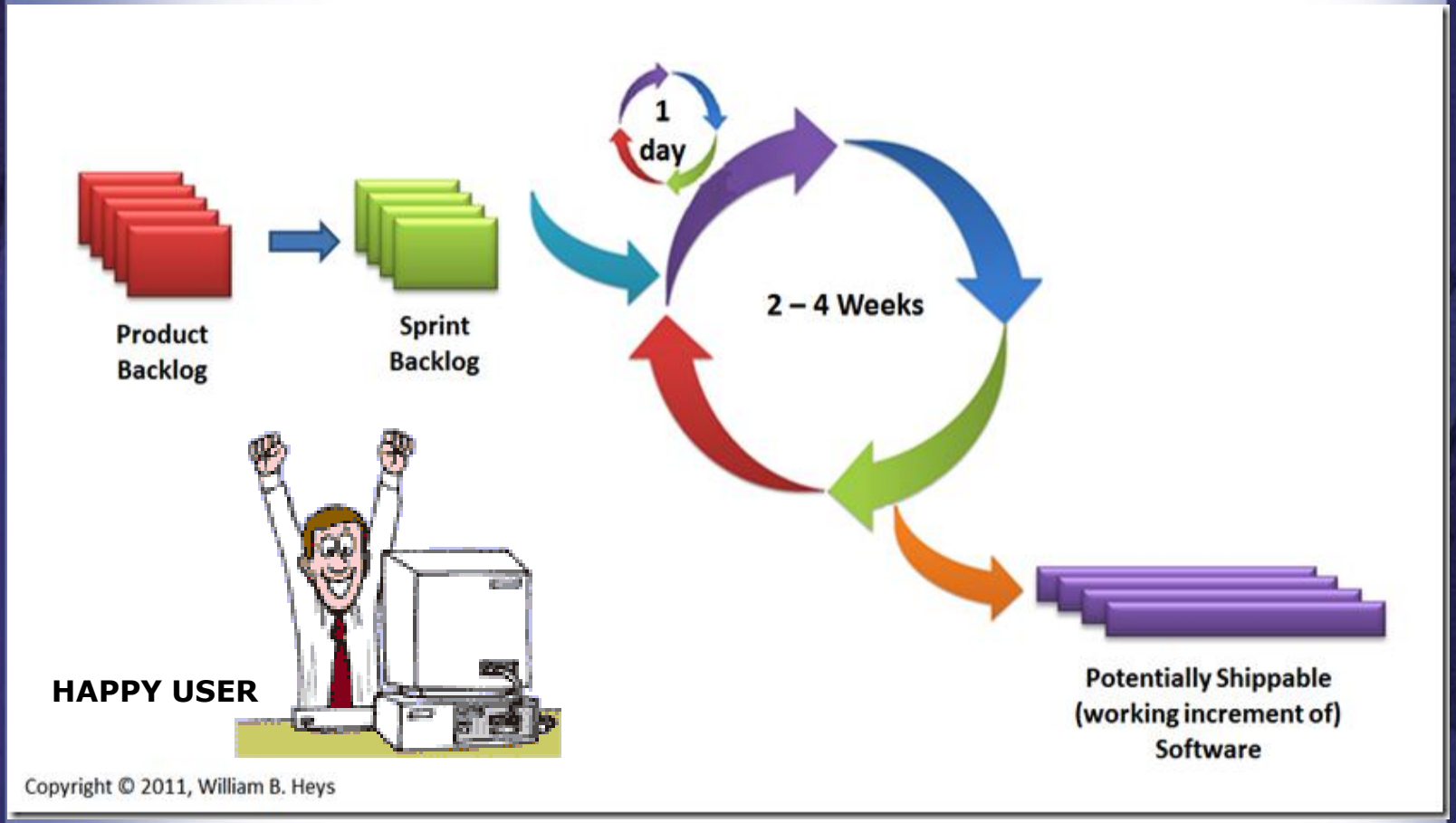
- Encourage customer collaboration

Lean-Agile (Scrum)

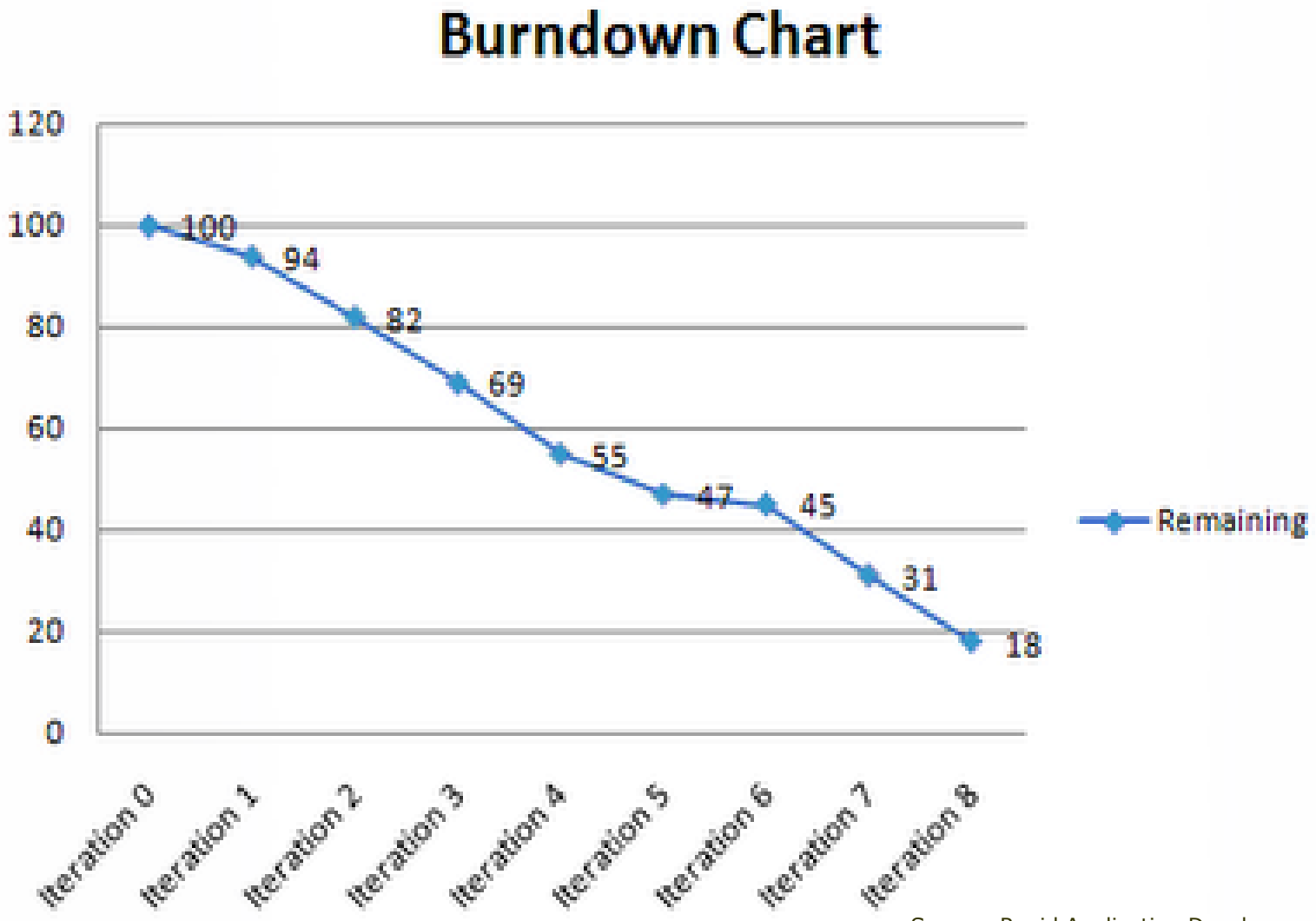
We need pigs on our team not chickens

Key principles of Scrum

- Eliminate waste
- Build quality in
- Defer commitment
- Deliver fast



Measure progress – burndown



Source: Rapid Application Development

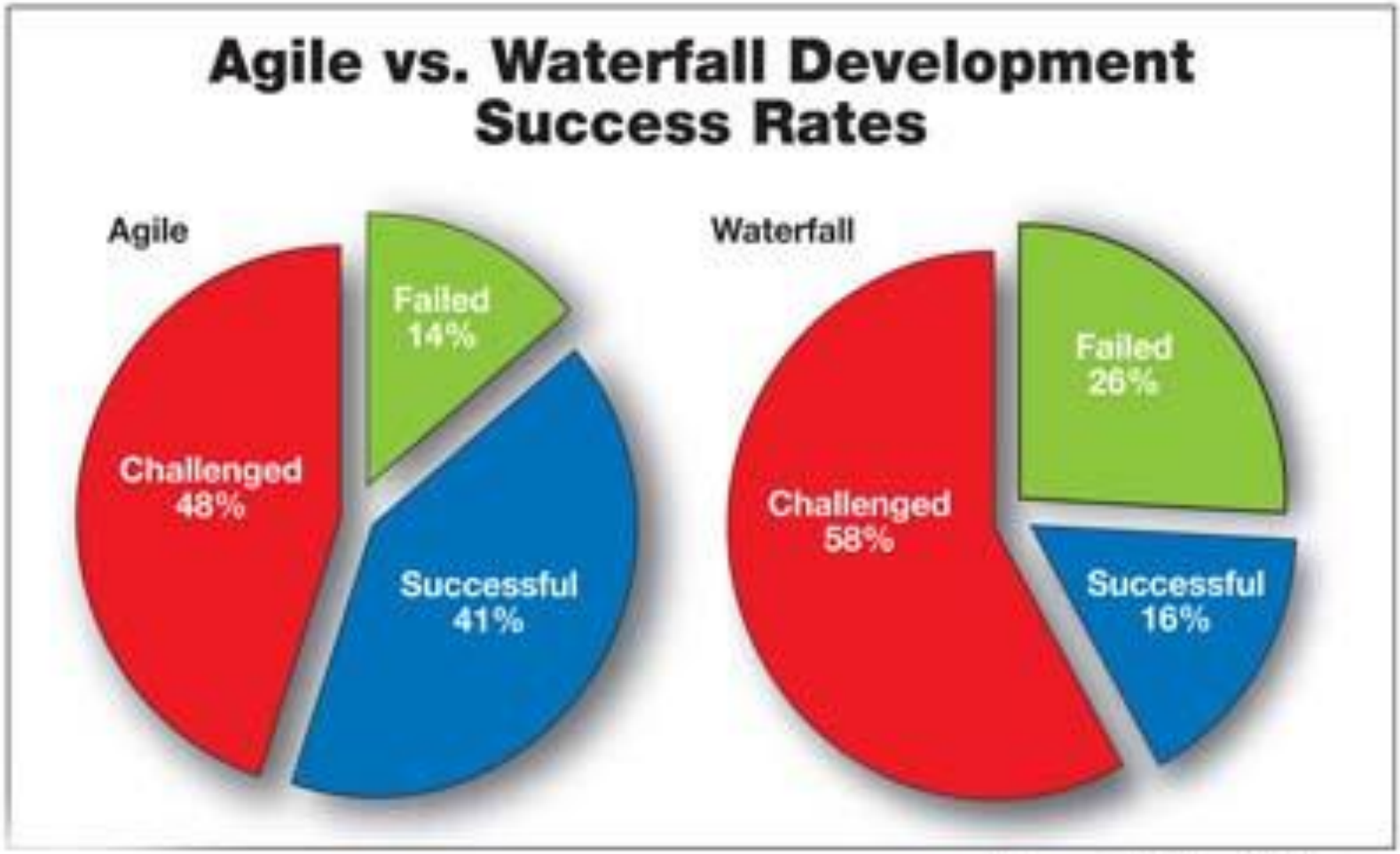
How have we managed risk?

- At the end of each iteration (2 to 4 weeks) we have a shippable product.
- It is an approach where capabilities are incrementally implemented in every successive iteration based on value to the user.
- At every iteration the work and the process are improved.
- Scrum is a starting point. There is no end state because of continuous improvement. The development stops when the user says the product is usable (or shippable).

Start a Lean-Agile software pilot in your company

- Clarify and define your values. You cannot lead if you don't know what you stand for.
- Ensure that the project is aligned with your organization's priorities and you have strong C-level support.
- Lean-Agile's success rate is much higher than traditional methods.

Lean-Agile guarantees success



Thank you.

Questions?

Dr. Samuel Prasad
Technology Management Consultant
631-368-8130

Email: samuel@prasads.com
Blog: blog.prasads.com