



# Training Solutions

## Agile Training Game

---

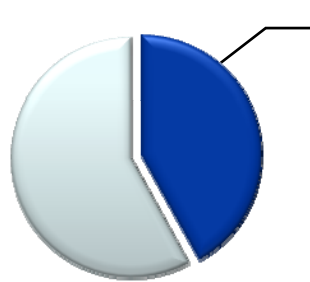
v. 7.3.1

---

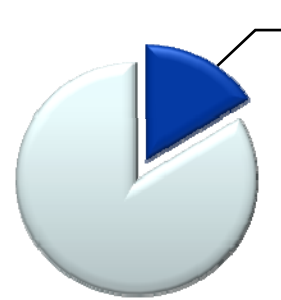
**Giora Morein**  
**Agile Coach and Trainer**  
**BigVisible Solutions**  
**gmorein@bigvisible.com**

### Some Interesting Findings:

A study of 8,300+ projects completed by large American companies using traditional methodologies revealed:



Only 42% of features originally proposed are implemented



Only 16% of projects finished on-time, on-budget with originally defined scope

### Most Common Reasons for Failure:

- ⇒ **Incomplete or changing requirements**
- ⇒ **Lack of User Input**

### **FACT!**

**80% of a Product's Features are added after the first release**

- ▶ Agile Software Development processes were first formulated in the early 1990's
- ▶ The Agile core set of values was articulated in the **Agile Manifesto** in 2001

*We Value:*

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

*“That is, while there is value in the items on the right, we value the items on the left more”*

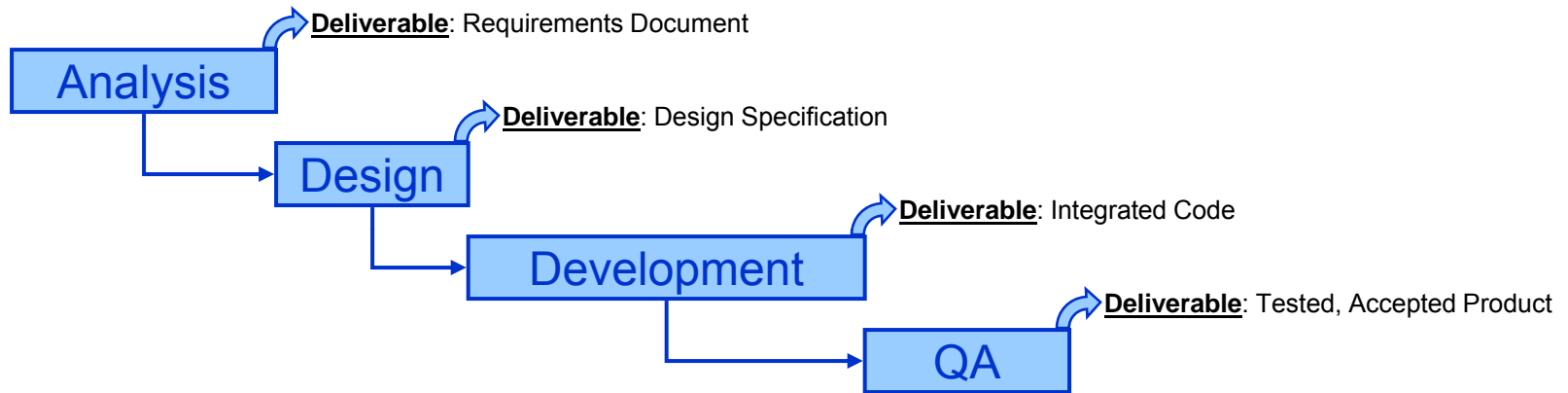
- ▶ Agile processes present light-weight methodologies that focus on incremental, iterative development
- ▶ Agile implementations focus on rapid value delivery, minimizing waste and continuous improvement

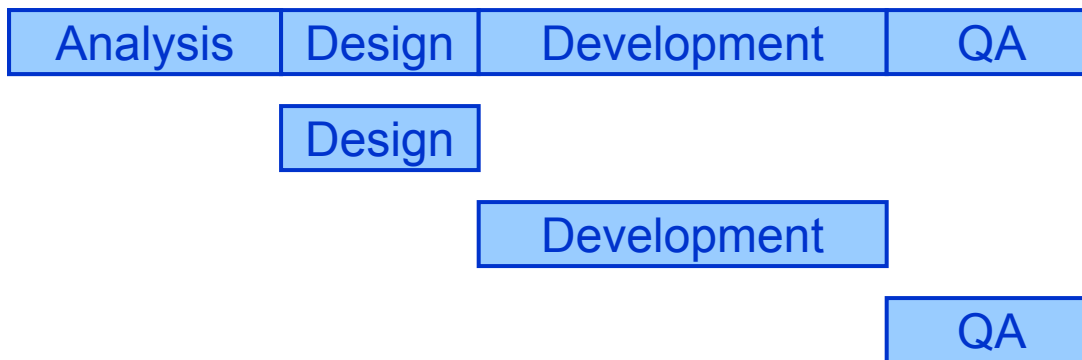
- ▶ **Faster Release Cycles**
  - Reduced time to market
- ▶ **Better Productivity**
  - Team focused on product development
- ▶ **Increased Quality**
  - Quality controls embedded throughout lifecycle
- ▶ **Focus on Delivering True Value**
  - Value can only be recognized in the form of completed product
- ▶ **Responsive to Change**
  - No change control process required

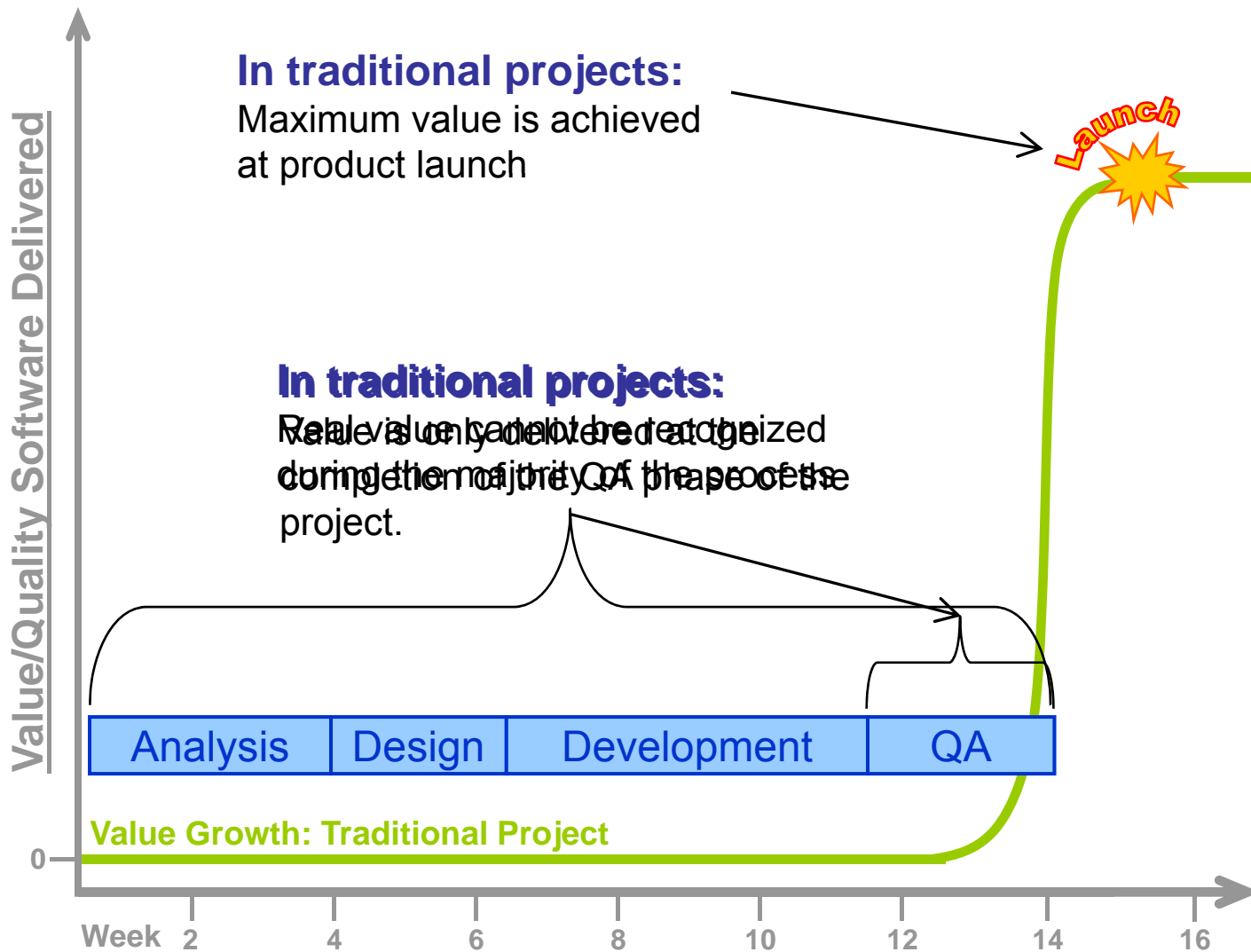
- ▶ **Focus on Business Objectives**
  - Deliver maximum business value
- ▶ **Measure Progress in the Form of Completed Product**
  - Tested, customer-accepted product
- ▶ **Build a Culture of Continuous Improvement**
  - Continuous incremental gains
- ▶ **Responsive to Feedback and Learning**
  - Facilitates changes that improve product
- ▶ **Maximize Team Productivity**
  - Laser-focus on delivering product

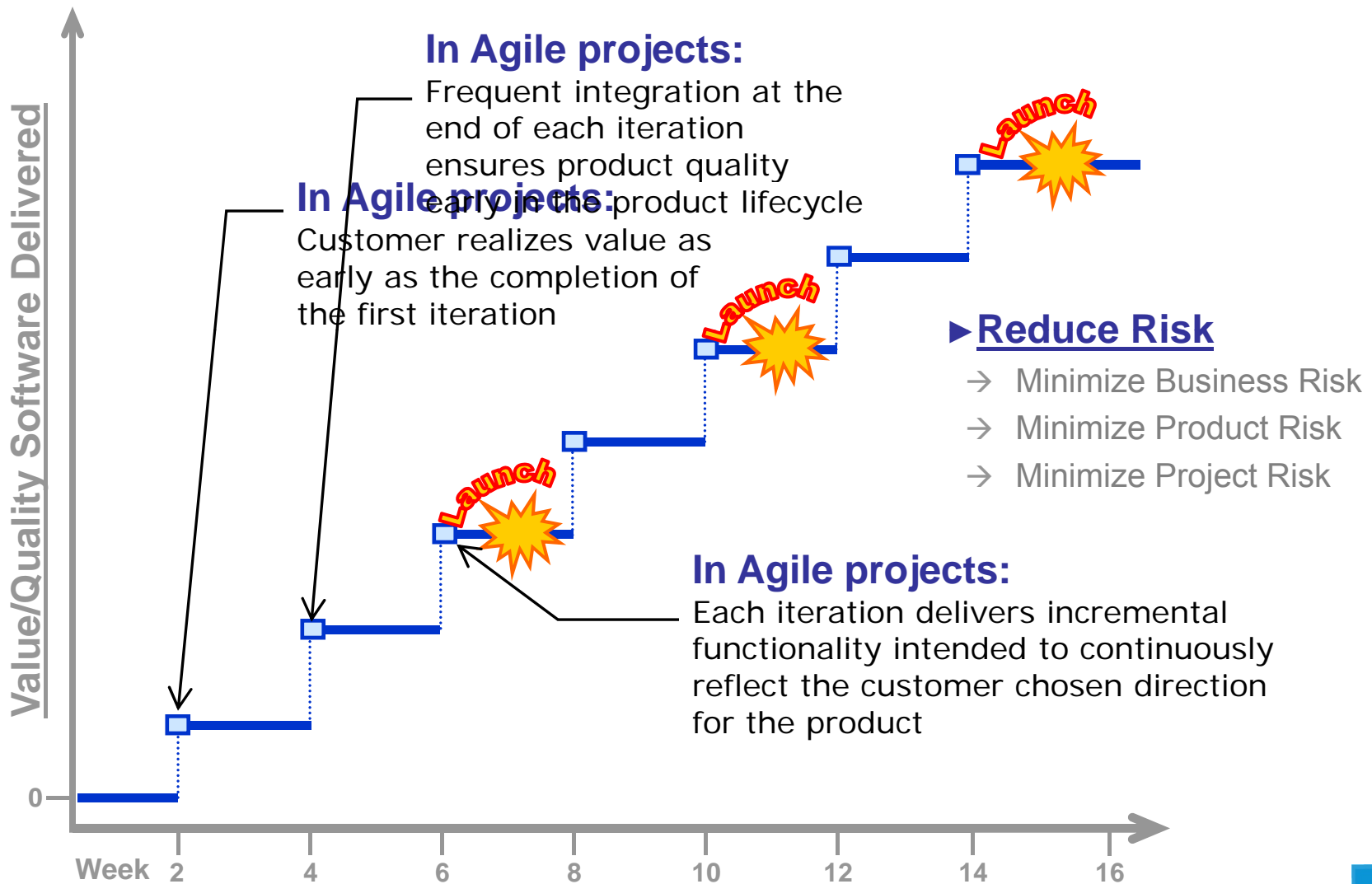
- ▶ Deliver customer value early and often
- ▶ Frequent, iterative delivery of accepted product
- ▶ Embrace change
- ▶ Closely integrated team
- ▶ Minimize waste
- ▶ Strive for highest level of collaboration
- ▶ Success is built around motivated self-organizing teams
- ▶ Continuously seek ways to improve

## Typical Waterfall/Traditional SDLC









- ▶ Deliver customer value early and often
- ▶ Frequent, iterative delivery of accepted product
- ▶ Embrace change
- ▶ Closely integrated team
- ▶ Minimize waste
- ▶ Strive for highest level of collaboration
- ▶ Success is built around motivated self-organizing teams
- ▶ Continuously seek ways to improve

## ▶ Agile Abandons all Rigor

→ Agile is very rigorous in its principles

## ▶ Agile does not scale

→ Agile has been scaled to hundreds of product developers

## ▶ Agile is not for Distributed Teams

→ Agile supports the model of multiple distributed teams that adopt the same principals and practices

## ▶ Agile = 'No Documentation'

→ Agile stresses product documentation

## ▶ Agile Focuses Cost Reduction

→ Agile does not directly address project or resource costs

## ▶ Agile Ignores Architecture

→ Agile stresses continuous architecture and design evolution

## ▶ Agile Means 'No Planning'

→ Agile stresses the importance of short-term and long-term planning