



## Introduction to Agile

Brian Swan  
[bswan@exoftware.com](mailto:bswan@exoftware.com)



# Requirements Change

## Requirements Change

Challenged Project Factors	% of Responses
Incomplete Requirements	12.3%
Changing Requirements	11.8%

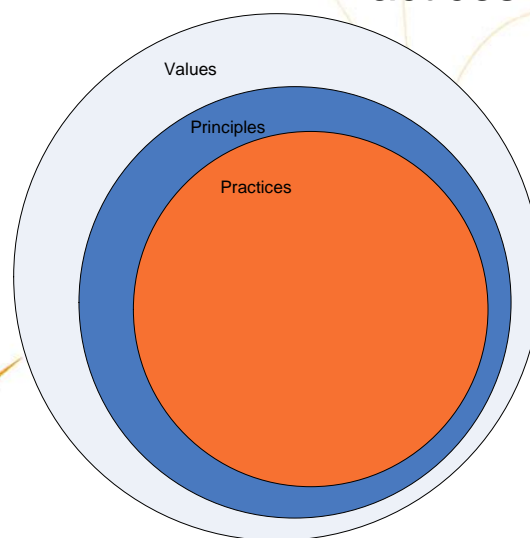
Impaired Project Factors	% of Responses
Incomplete Requirements	13.1%
Changing Requirements	8.7%

*The Chaos Report, Standish Group, 1995*

## Agile Family

- eXtreme Programming (XP)
- SCRUM
- Adaptive Software Development
- Lean Development
- Feature Driven Development
- Crystal
- DSDM

## Values, Principles & Practices



## Values - Agile Manifesto

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

## Principles

- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- Business people and developers must work together daily throughout the project.

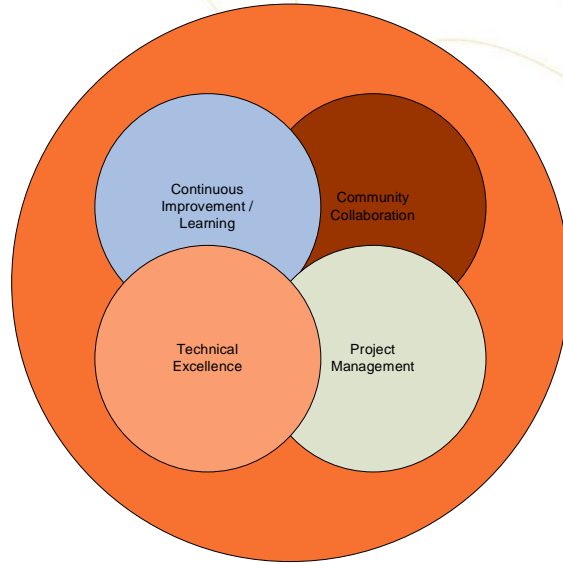
## Principles

- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- Working software is the primary measure of progress.
- Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

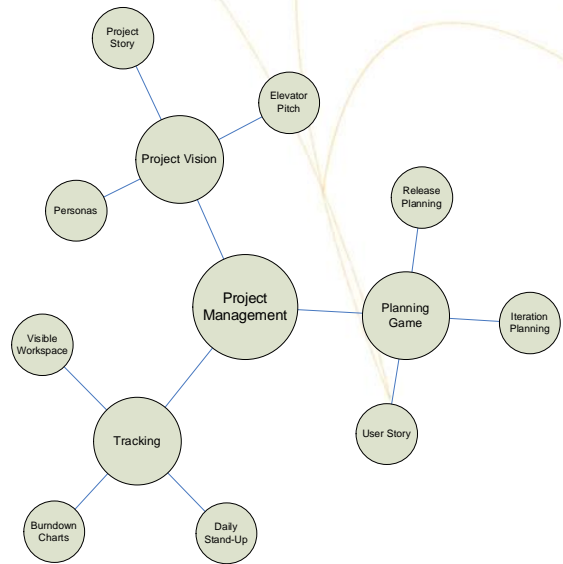
## Principles

- Continuous attention to technical excellence and good design enhances agility.
- Simplicity--the art of maximizing the amount of work not done--is essential.
- The best architectures, requirements, and designs emerge from self-organizing teams.
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

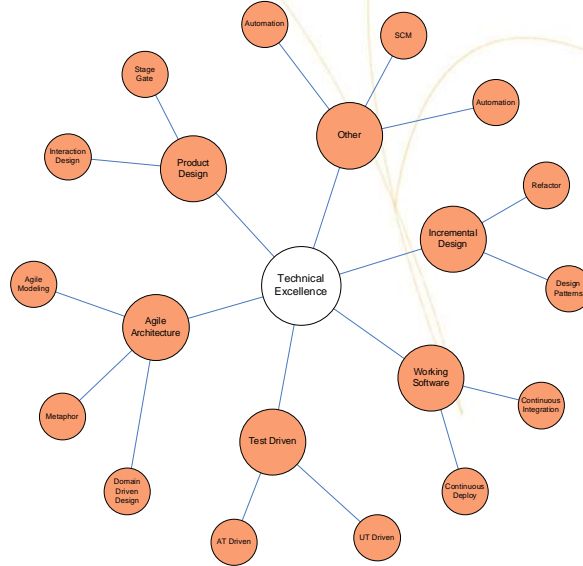
## Practices



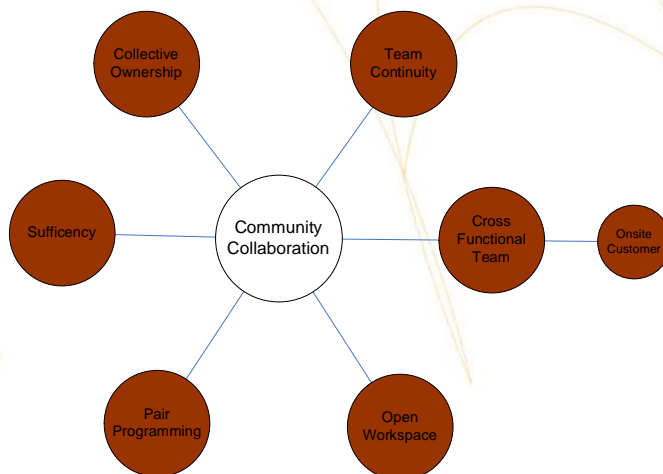
## Project Management



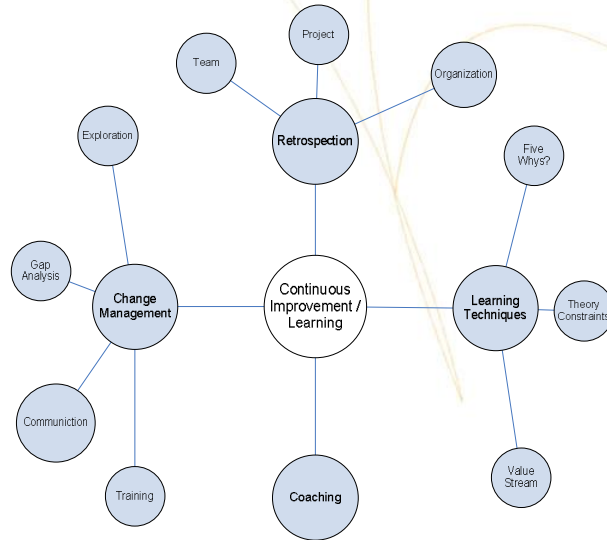
## Technical Excellence



## Collaboration



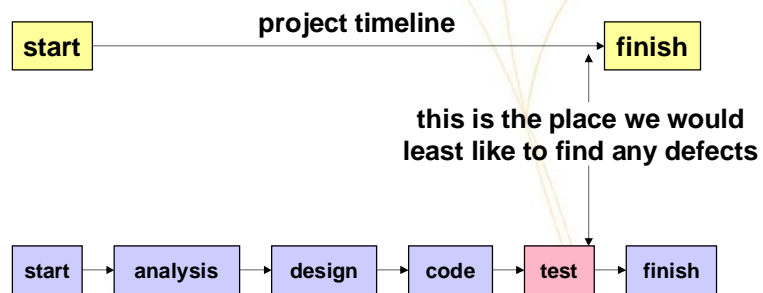
# Continuous Improvement



## Why Agile?

- Quality
- Predictability
- Productivity

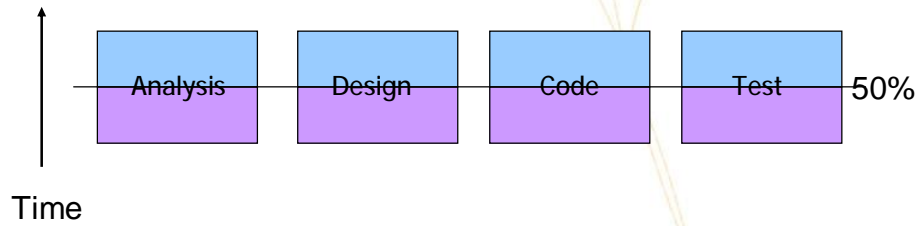
## Traditional Testing



the design of the waterfall means that we will always be at our **most vulnerable** just when we least want to be and that defects will be found where they are the **most expensive** to fix.



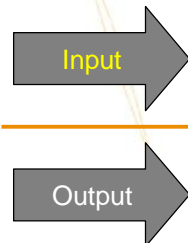
## Predictability - Invert the axis



## Productivity



Productivity =

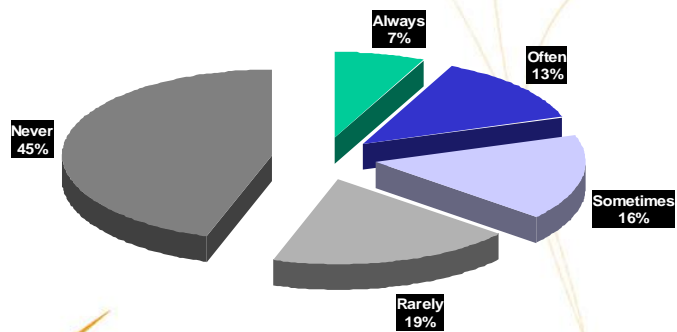


## Increase Productivity

Do Less Work

Produce More Value

## Do Less Work



...identify these high value features and  
prioritize them appropriately...

## Productivity - Small Is Beautiful

Project Size	People	Duration (Months)	Success Rate
Less than \$750K	6	6	55%
\$750K - \$1.5M	12	9	33%
\$1.5M - \$3M	25	12	25%
\$3M - \$6M	40	18	15%
\$6M - \$10M	+250	+24	8%
Over \$10M	+500	+36	0%

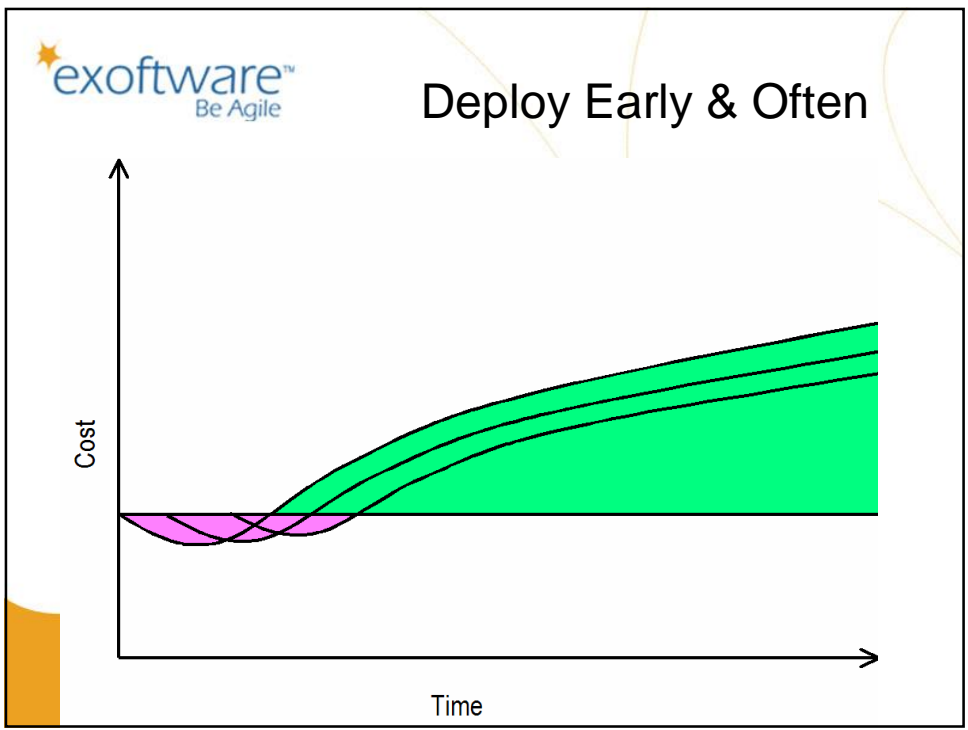
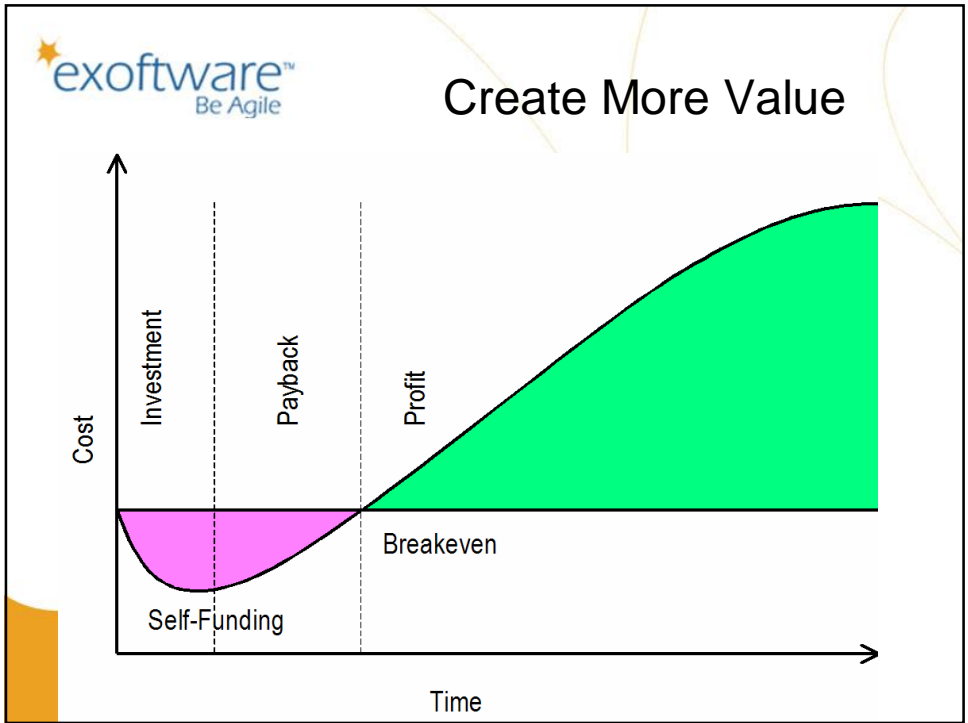
*CHAOS, The Standish Group, 1999*

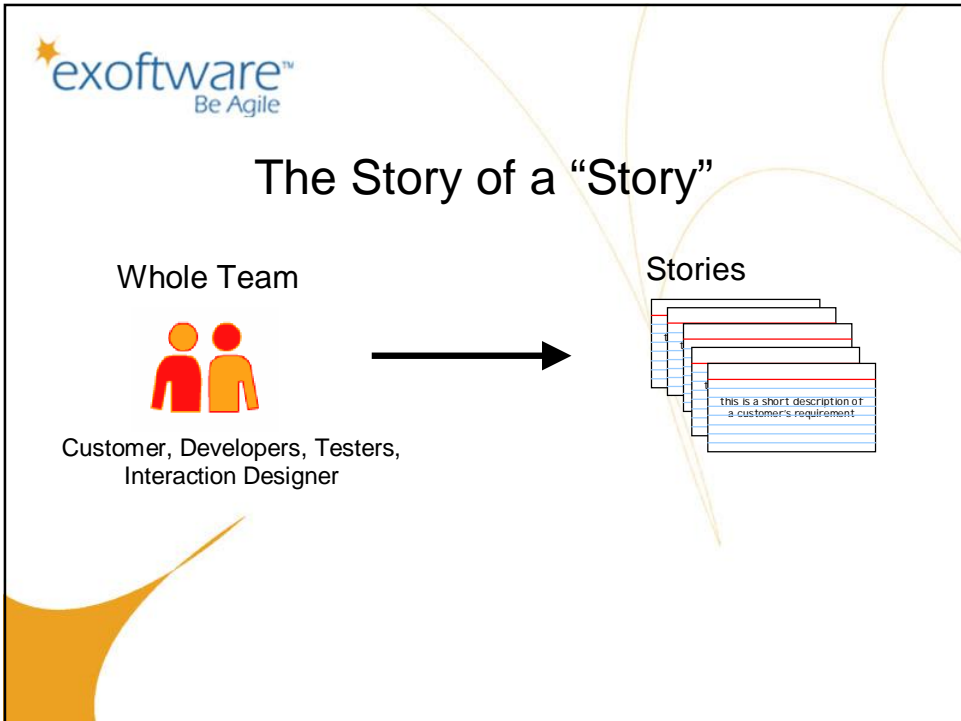
## Productivity - Small Teams

QPW had a small core team--four people--who interacted intensely over two years to produce the bulk of the product.

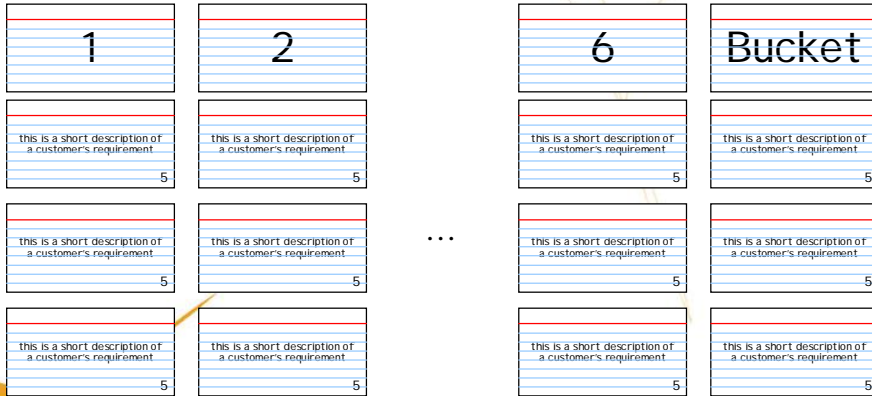
A million lines of code were written over 31 months by about eight people.

*Borland Software Craftsmanship: A New Look at Process, Quality and Productivity, James O. Coplien, 1994*

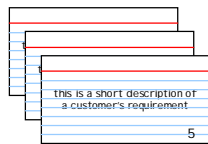




## Release Planning



## Iteration Planning



+

Automated  
Acceptance Tests



# Acceptance Tests

```

StartApplication

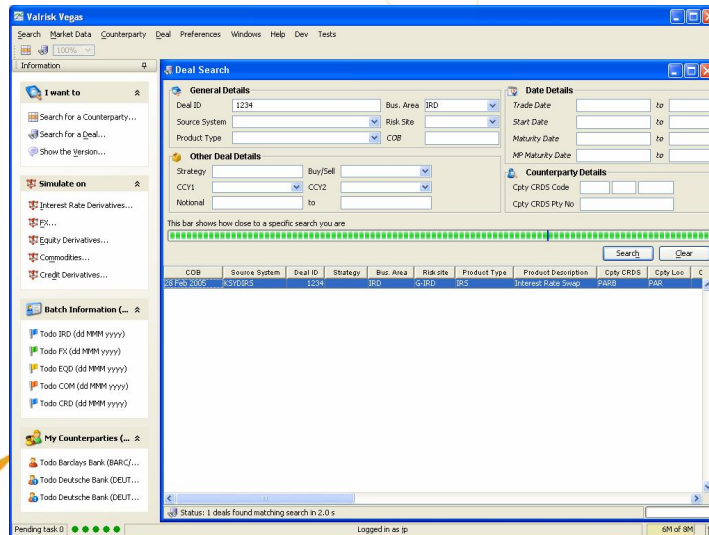
# Search with no deals returned
ClickMenu      SearchDeal
EnterText      DealId      -1
ClickButton    FindDeal
CheckTableRowCount SearchDealResults 0
CheckLabel     StatusBar    "Status: 0 deals found matching
  search"

# Ensure cannot search without at least 1 character
EnterText      DealId      ""
CheckEnabled   FindDeal    false
EnterText      DealId      "1"
CheckEnabled   FindDeal    true

# Ensure we cannot search by entering spaces
EnterText      DealId      " "
CheckEnabled   FindDeal    false

# Ensure Single external deal id
EnterText      DealId      1234
EnterText      TradeDateFrom ""
EnterText      TradeDateTo  ""
ClickButton    FindDeal
CheckTableRowCount SearchDealResults 1

CloseApplication
  
```





## Succeeding With Agile



## Success Factors

- Enthusiastic development team.
- Committed customer.
- Knowledgeable on-site QA resource.
- Didn't cherry pick practices.
- Open workspace.
- Regular retrospectives.

## No Silver Bullet

- Good software is written by good people.
- The practices and techniques are learnable.
- Can't force people to use Agile.
  - too many practices can be “faked”.
- Agile methods recognise reality.
  - Involve the customer
  - Deal with change
  - Can't get it right first time
  - Let good people write better software by getting out of the way.