

**AGILE AND AGILE TESTING**

**KAIZANIA**

© Kaizania 2009



**9 DECEMBER 2009**

**Lionel Bisschoff / Arrie van der Dussen**

**Changing the way software products and services  
are created**

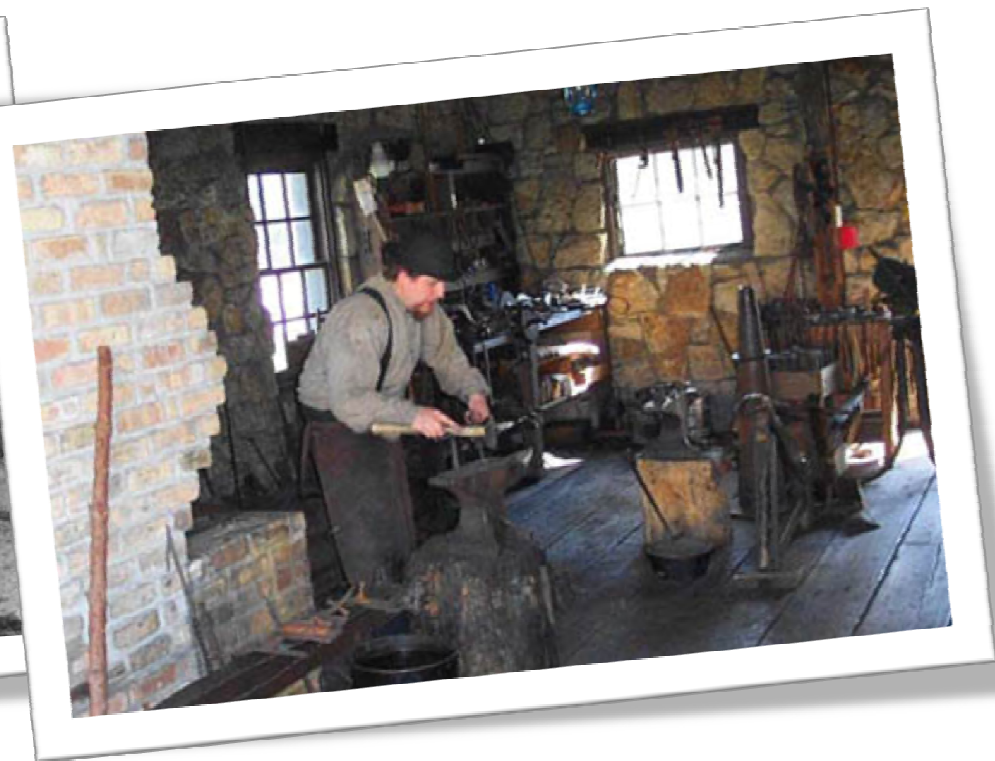
## **flexible production era**

-----  
impact on software development

three eras of work

---

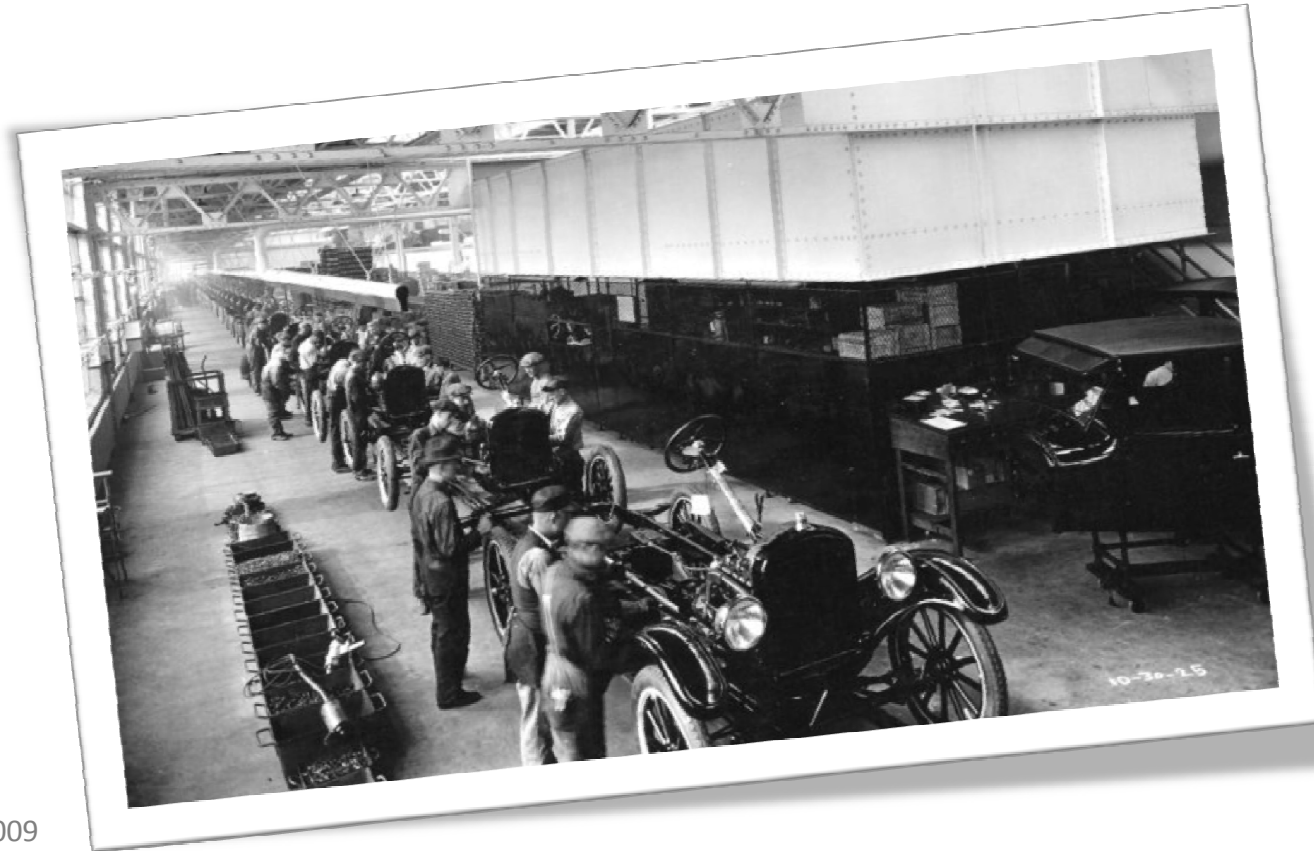
## CRAFT Production: past -> 1920's



three eras of work

---

## MASS Production: 1920's -> 1950's



## mass production era

---

### craftsmanship replaced through

- craft knowledge systematically collected
  - operations simplified into consecutive steps
  - steps specified in great detail
  - work stations designed to execute steps
  - simplification and standardisation of tasks
  - workers rigidly supervised
  - must complete tasks with no deviation, no input
- 
- intensified division of labour
  - 1000's of workers
  - increasing volume of information to be tracked
  - large management structures to cope

1920's



1950's

## mass production era

---

### influence on project management?

- Project plans
- Gantt charts
- Work breakdown structures
- Standardisation
- Repeatability
- Process

- CMMi
- Prince II
- PMBOK

1920's

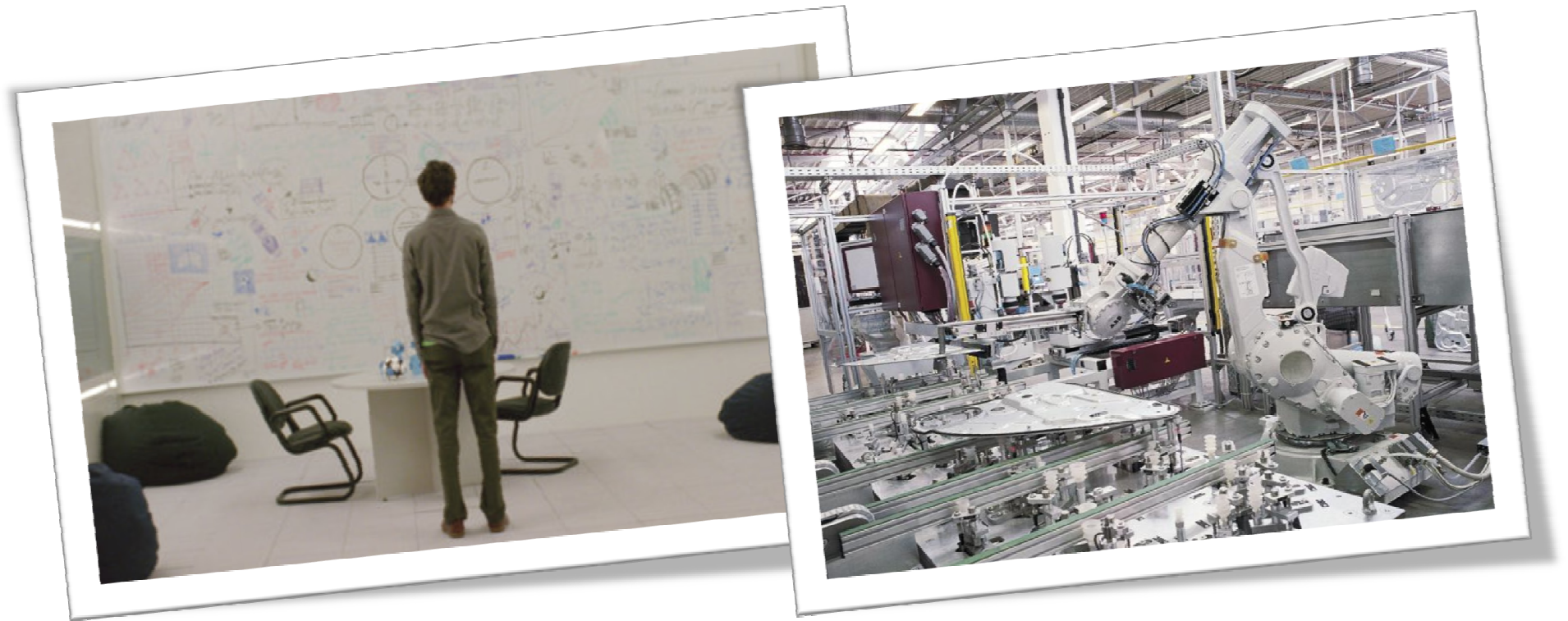


1950's

three eras of work

---

## FLEXIBLE Production: 1980's -> now, ongoing

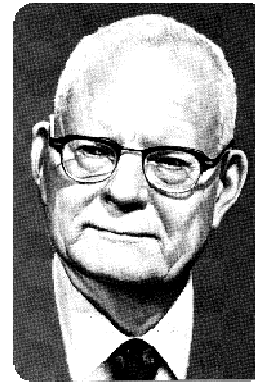


## flexible production era

---

### W.Edwards Deming, Walter Shewhart

Statistical Process Control, Quality control  
Continuous Improvement, Kaizen



### Toyota, Motorola

The Toyota Way, Six Sigma, JIT, Lean,  
New New product development game



1950's



today



## flexible production era

---

### Quality

$$\text{quality} = \frac{\text{business value delivered}}{\text{total cost}}$$

**NO TRADE-OFF**

**Higher quality drives Increased Production AND  
Lowers Costs**

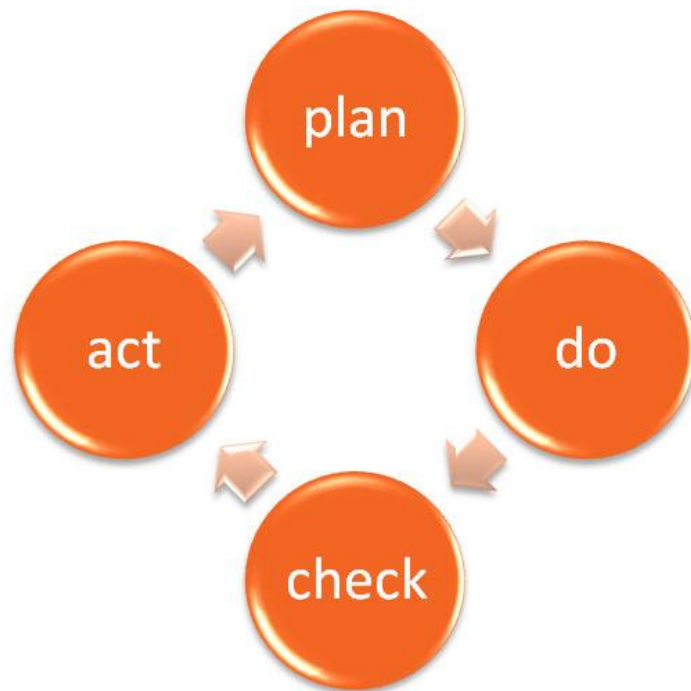
***All non value adding activities from the CUSTOMER  
perspective are WASTE***

1950's



today

## Continuous Improvement - Kaizen



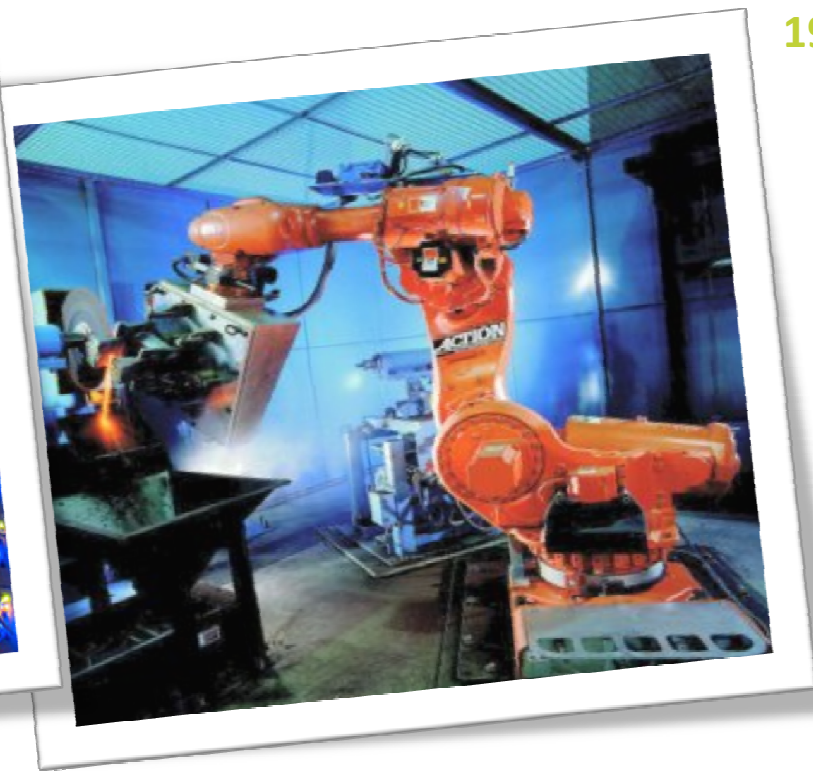
- Inspect
  - Propose improvement
  - Try
  - Repeat
- 
- Improve Quality
  - Increase Production
  - Reduce Costs
  - Maximise Customer Value

1950's



today

## Technology



1950's



today

- Store and retrieve vast amounts of information
- Automate computation
- Automate manufacturing, production

# Flexible Production Era

## Product Development



Replace

- Serialized, Independent phases
- Independent, single skill teams
- Management co-ordination



With

- Overlapping development phases
- Cross functional teams
- Self-organizing

1950's



today

flexible production era

---

## Influence on Software Development?

Ken Schwaber, Jeff Sutherland



Kent Beck



UK DSDM Consortium



I. Jacobson, G. Booch



R. Charette, M. / T. Poppendieck

Lean Software Development

1990



2001

## agile manifesto

---

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to 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.

Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland



## agile development

---

### Accept that:

- Software Product development is repeated invention
- Perfectly planned invention is not possible

### What works?

### Vision realised through

idea, try, test, learn, more work needed  
idea, try, test, learn, more work needed  
idea, try, test, learn, more work needed  
idea, try, test, perfect, done

Iterations, iterations, iterations, iterations, iterations, iterations,  
iterations, iterations, iterations, iterations, iterations, iterations

## What works?

---

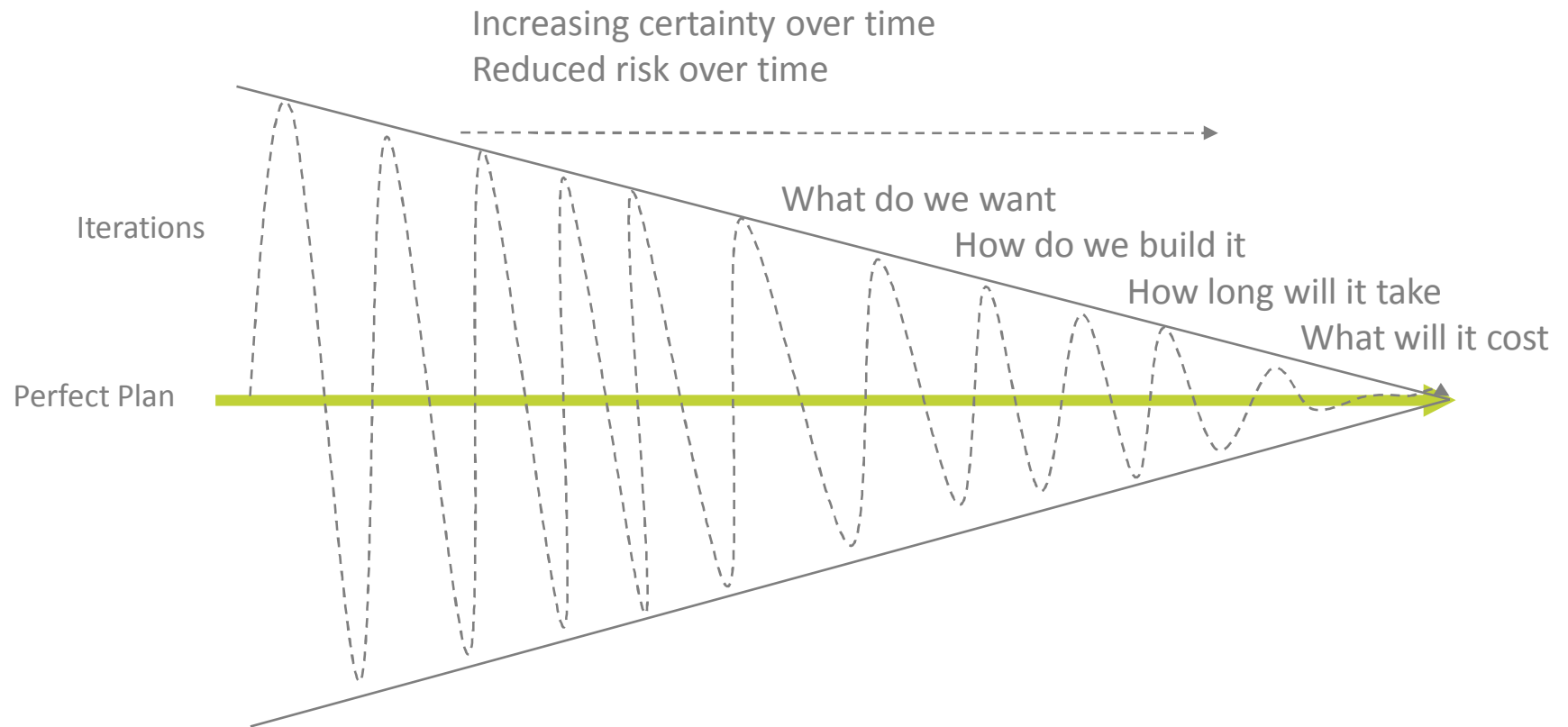
### Invent software through:

- Empirical process control
- Iterations and continued learning
- Goal driven self-organizing teams



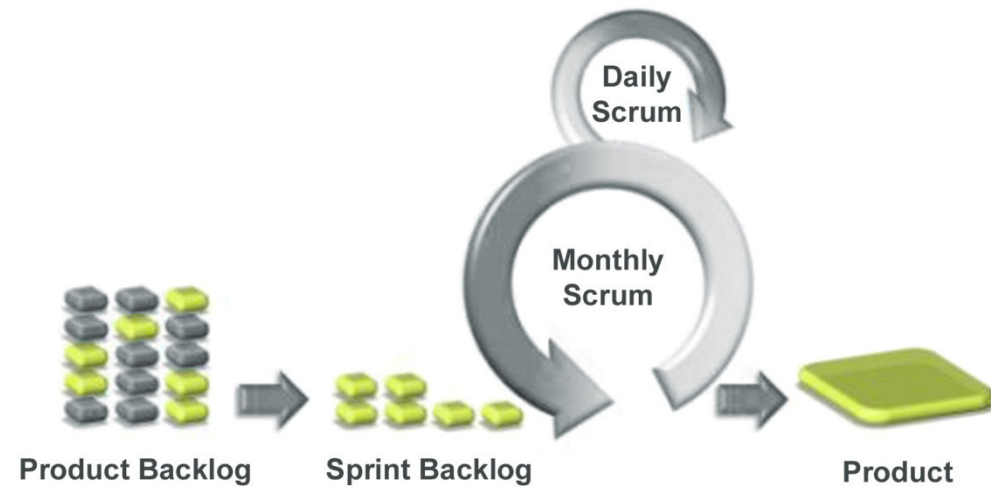
## perfect plan vs reality

---



## iterations change everything

---



### Iterating well means:

- JIT Requirements
- Fixed iterations, variable scope
- Adapting/learning fast vs. conformance to plan
- Coding/refactoring, integrating, testing, accepting repeatedly
- Automation: Coding Standards, static analysis, dependency management, builds, testing
- Cross functional, dedicated teams
- Adaptive leadership, collaboration, trust, ownership, light touch
- Self organisation
- Organisational structure geared to maximising ROI from iterative invention

## agile : the difference

---

<b>Mass Production</b>	<b>Flexible Production</b>
Predictive long term detail planning	Constant adaptive planning
Top-down control	Mix of top-down control and self-organizing teams
Manager as thinker	Manager-as-coordinator
Workers as implementers	Workers as thinkers/implementers
Line vs. staff – thinkers separated from doers	Thinkers as doers
Division of labour	Accepted responsibility / volunteerism
Exhaustive Requirements gathering	Product Vision, Release planning, Just in Time Requirements
Commands + Manage and Control	Commitments + Leadership

# Agile Testing

---

## agile manifesto

---

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to 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

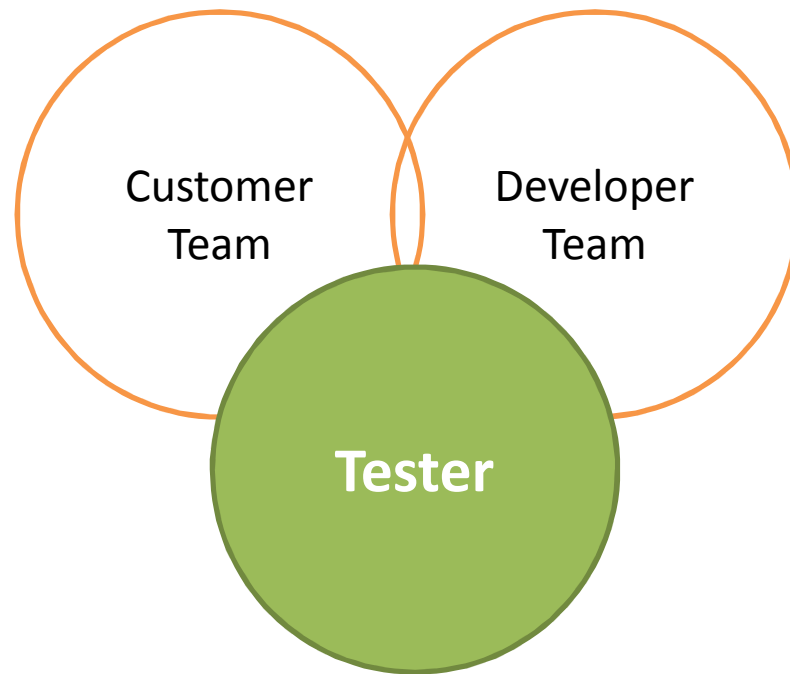
That is, while there is value in the items on the right, we value the items on the left more.

Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland

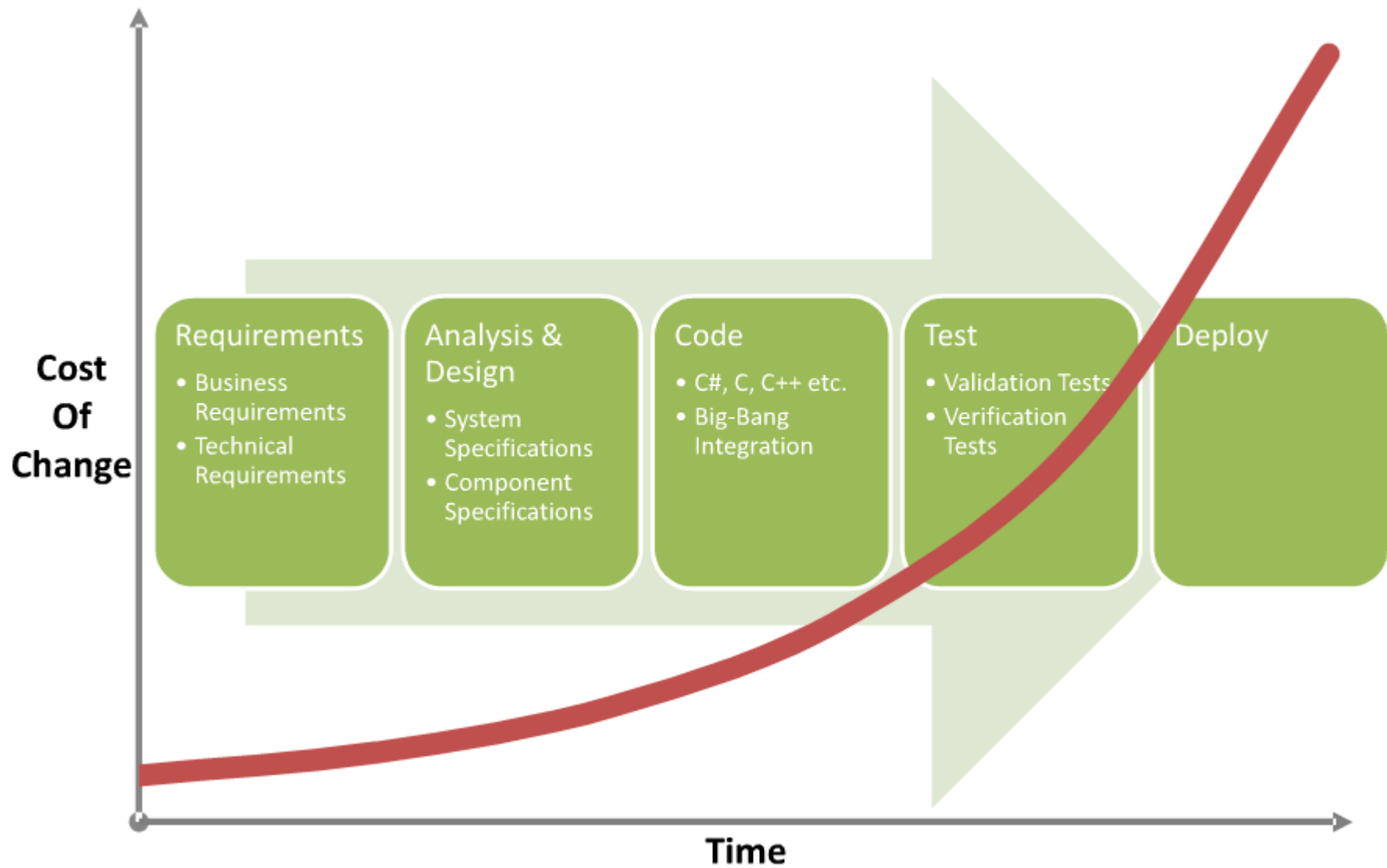


## Roles & Activities on an Agile Team

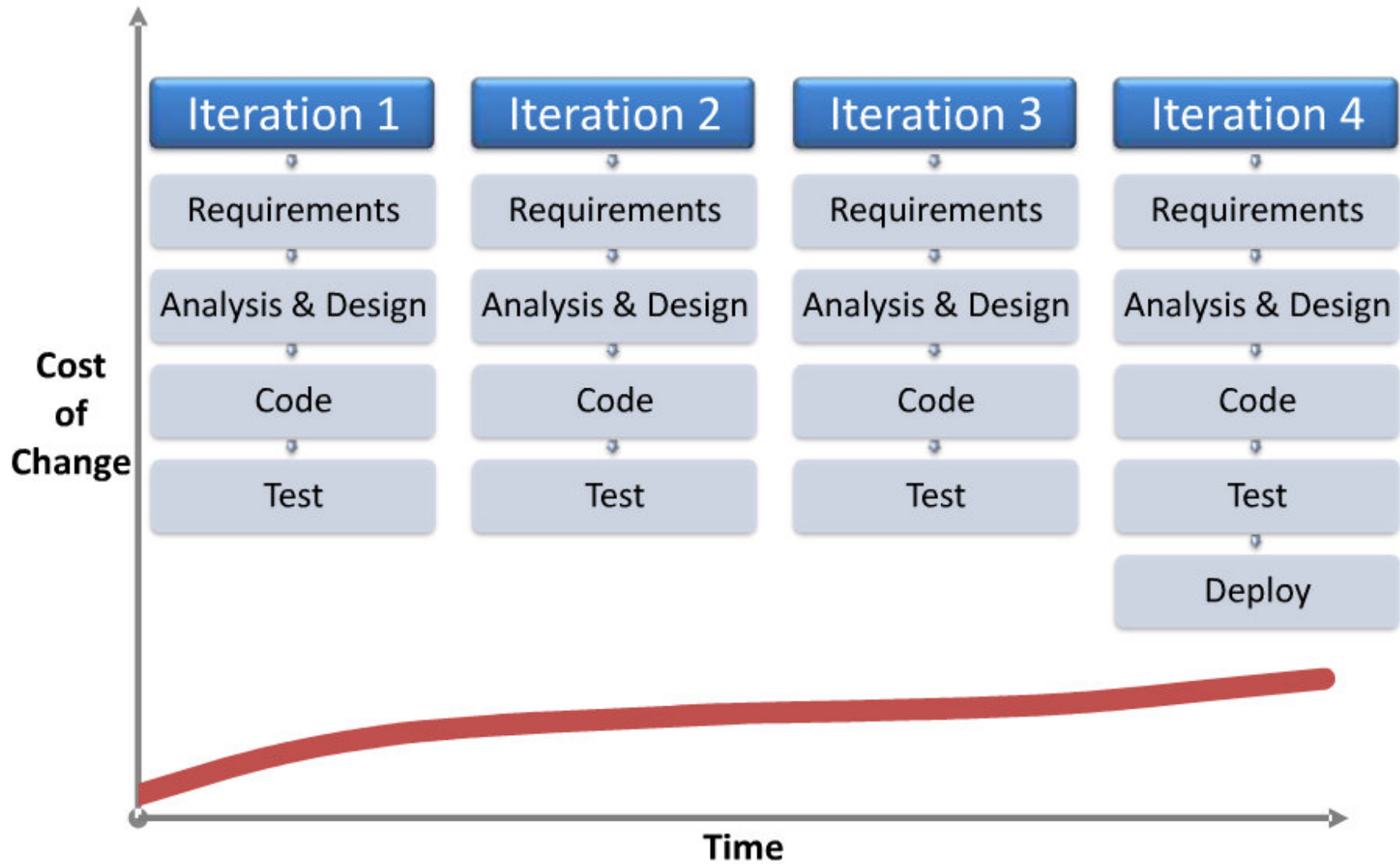
---



## Traditional Testing



## Agile Testing





## Traditional vs Agile Testing

---

Traditional testers confirm a set of requirements, where Agile testers confirm business value delivered.

## Whole Team Approach



## Agile Requirements - Stories

As a <role>, I want to <feature>, so that <value>

**I**ndependent

**N**egotiable

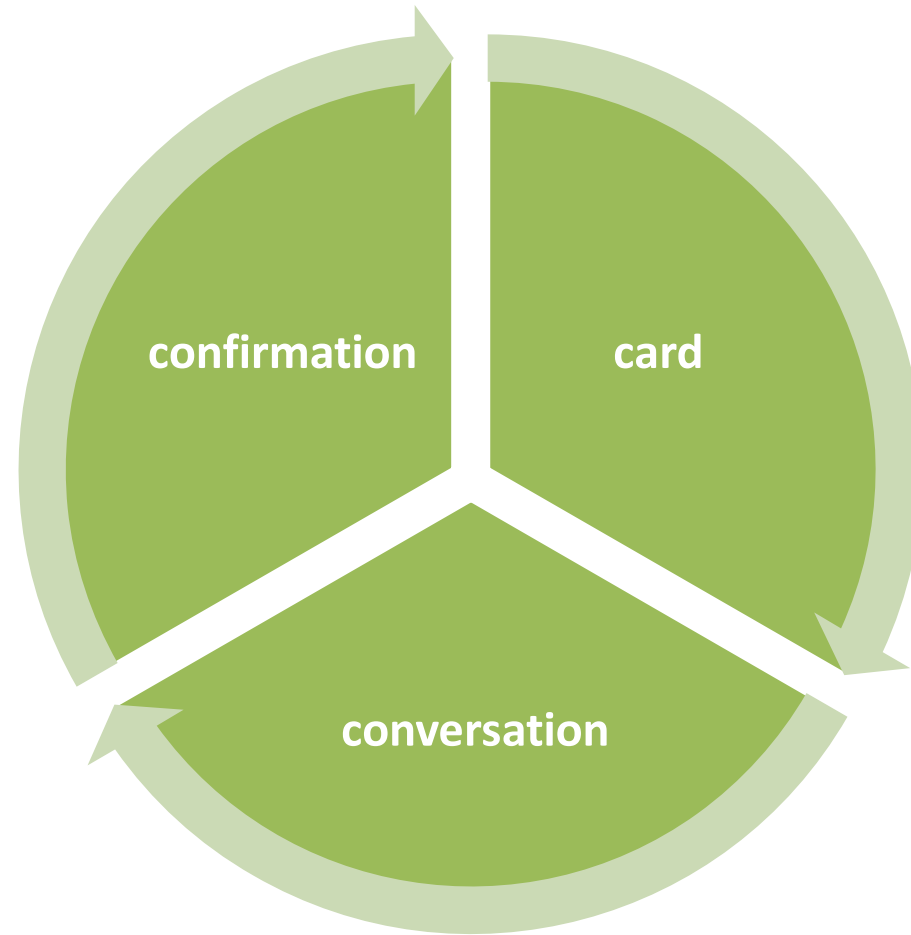
**V**aluable

**E**stimable

**S**mall

**T**estable

## Quality driven requirements

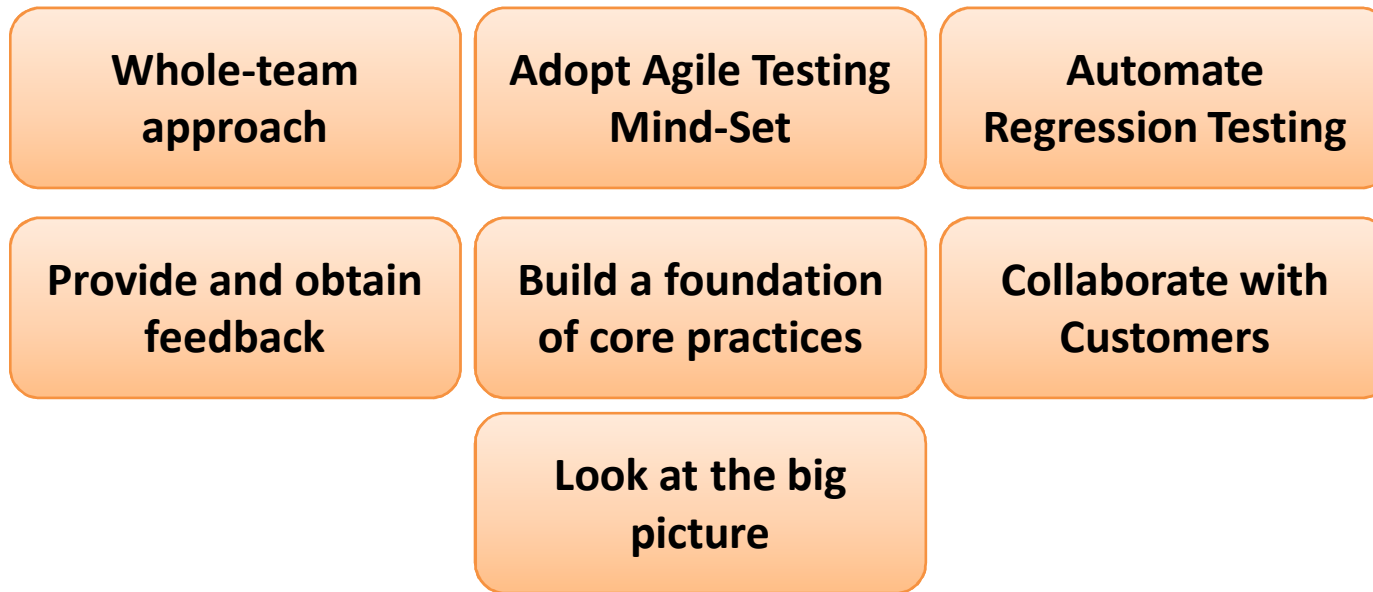


# WHAT IS AN AGILE TESTER?

**‘A professional tester who embraces change, collaborates well with both technical and business people, and understands the concept of using tests to document requirements and drive development’**

- Lisa Crispin ( Agile Testing )

# KEY SUCCESS FACTORS TO AGILE TESTING



- Lisa Crispin ( Agile Testing )

# XP Tester's Bill of Rights

*from Testing Extreme Programming by Lisa Crispin and Tip House (Addison-Wesley, 2002)*

You have the right to bring up issues related to quality and process at any time.

You have the right to ask questions of customers and programmers and receive timely answers.

You have the right to ask for and receive help from anyone on the project team, including programmers, managers and customers.

You have the right to make and update your own estimates for your own tasks and have these included in estimates for stories.

You have the right to the tools you need to do your job in a timely manner.

You have the right to expect your project team, not just yourself, to be responsible for quality.







Thank you...

[www.kaizania.co.za](http://www.kaizania.co.za)  
[thinkingagile.wordpress.com](http://thinkingagile.wordpress.com)

[agile@kaizania.co.za](mailto:agile@kaizania.co.za)  
[info@kaizania.co.za](mailto:info@kaizania.co.za)  
+27 83 700 2181