Ground System Architectures Workshop

"Doing More with Less"

Sheraton Gateway Hotel Los Angeles

March 18-21, 2013

NORTHROP GRUMMAN

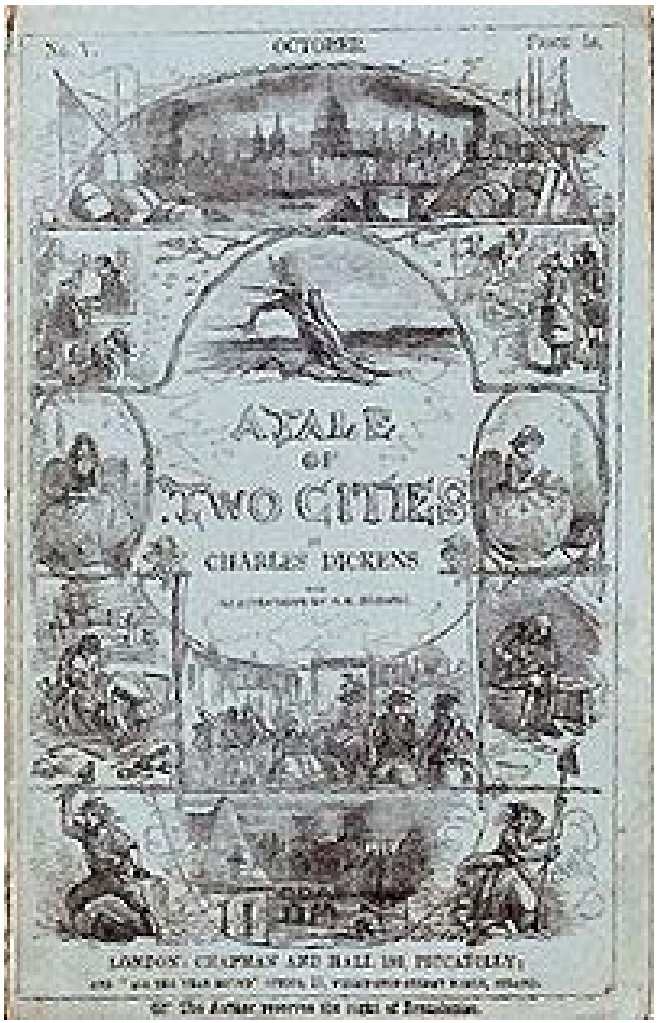
# Agile Versus Traditional – A Tale of Two Methodologies

March, 2013

Bret Rudnick

# A presentation told in chapters

- Analogous to the book in that:
  - Separate but related storylines
  - Each has a distinct focus
  - They intersect in various ways
  - Ultimately, it makes sense
- Different to the book in that:
  - No accusations
  - No  French idioms
  - Almost no violence




*It will all tie together and make sense*

# Presentation Agenda

- Book the First -- Traditional Methodology
- Book the Second -- Agile Methodology
- Book the Third -- Conclusions

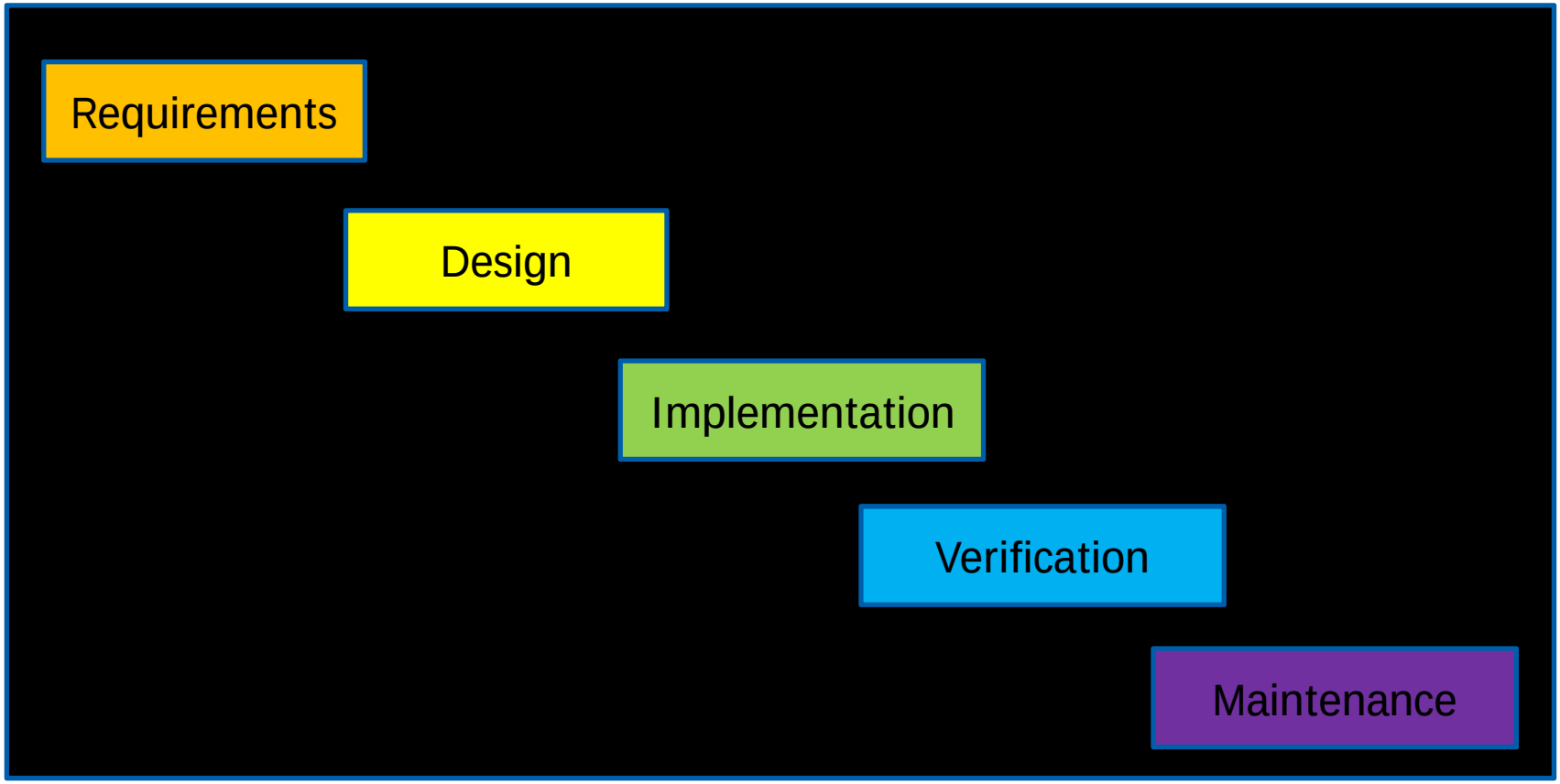
- **Traditional Methodology**

# Traditional Methodology

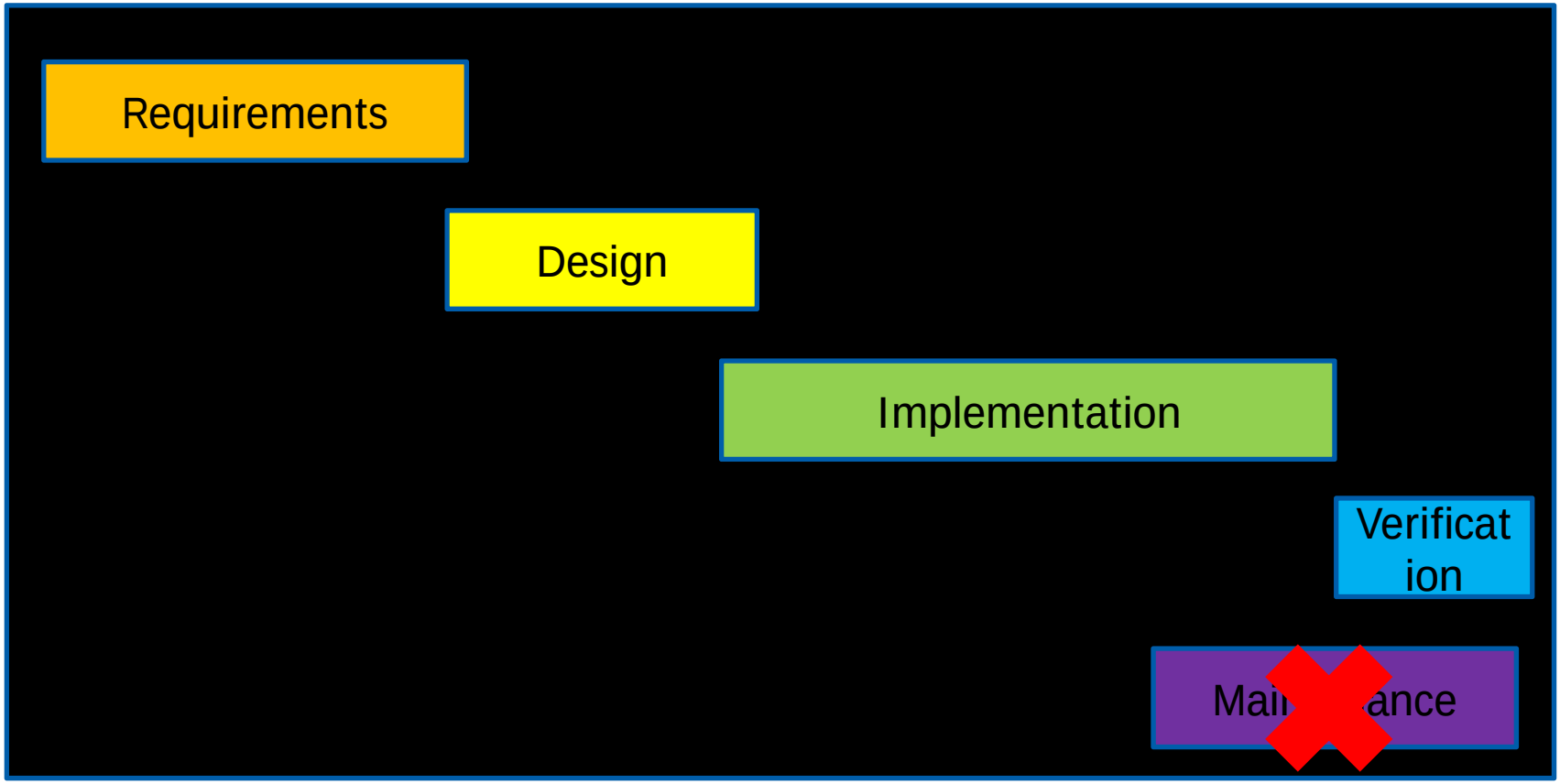



- Several methods but common phases
- Requirements definition
- Design Phase
- Coding Phase
- Testing and Requirements signoff

# Waterfall Development – Most Common



# Waterfall Method – Perhaps You've Seen This?

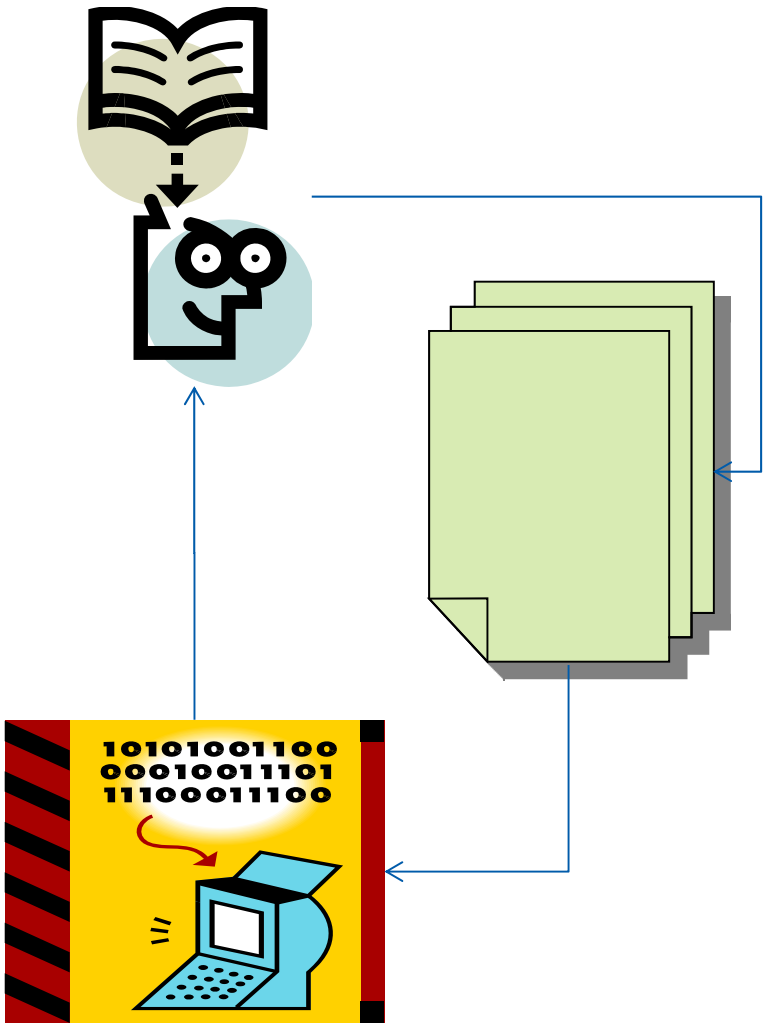


# Traditional Development – Advantages/Disadvantages

| Advantage | Disadvantage |
| --- | --- |
| Thorough requirements definition | Lack of flexibility for change |
| Design proven | At expense of other areas |
| Documentation emphasised | Less opportunity for innovation |
| Planning details | Test compressed |
| Known quantities | Change inhibited |

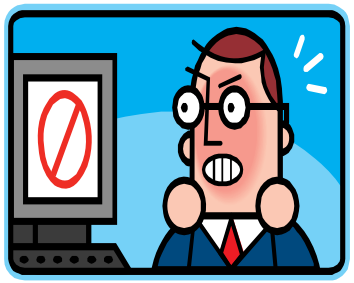
# Traditional Requirements Definition

- Customer defines requirements
- Developer works from (mostly) static list
- Customer sees final result



# Traditional Requirements Definition Outdated

- End result may differ from need
- Time lag between definition and results excessive
  - Typical timeline of requirements to delivery can be months or even years
- Requirements may pass acceptance testing, but:
  - Operational consideration may be absent
  - Actual use may highlight disadvantages
  - Pieces may not fit together optimally
  - Design may not allow for modification



A more collaborative approach is needed

- **Agile Methodology**

# Why is it important to develop in Agile?




"**The U.S. Government and the Department of Defense in particular, is now more than ever committed to the adoption and use of Agile methods when acquiring IT-intensive systems.**"

Dr. David F. Rico, "AFEI DoD Agile Development Conference", December 23, 2010, http://www.scrumalliance.org/articles/307

# What Makes an Agile Product?

- Meets all the requirements of a product/process to be Agile
  - Product and developing process respond to change without breaking the system
  - Modules can be mixed, reused, scaled, and (re)configured as required
  - Thought of future enhancements and the changing nature of the environment it operates in have also been considered
  - Close collaboration with the user/customer is also taken into account to facilitate change
  - Knowledge is managed

**Product and Process Can Be Declared Agile**

# Response Situation (RS) Analysis

| Change Domain | | General Issues |
|---|---|---|
| Proactive | Creation (and Elimination) | • Create modules to support different data tables {s}<br>• Capability to handle new data formats and modified data formats {s}<br>• Allow for additional modules to support enhancements{s} |
| | Improvement | • Positioning of data output for faster pattern recognition {q}<br>• Combine diverse real time and non-real-time data {q}<br>• Include table data from terminals {s} |
| | Migration | • Allow modules to include future assets (follow-on systems) {s}<br>• Changes in communication infrastructure {s, t}<br>• Allow users at different levels of security (e.g., International Partners) {s, t} |
| | Modification (Add/Sub Capability) | • Support new and existing user platforms {c, t}<br>• Support different COTS (or other) display tools {s, c, t}<br>• Allow room for additional/future applications {s,c} |
| Reactive | Correction | • Troubleshooting data processing {t, q}<br>• Verify with customer initial data set is, in fact, the one intended to process {q} |
| | Variation | • Modify data formats (size, contents, etc.) {s}<br>• Verify cross-constellations tables {s} |
| | Expansion (and Contraction of Capacity) | • Add new data tables {s}<br>• Remove old data tables as satellites are retired {t} |
| | Reconfigu-ration | • Reconfigure the toolset to meet changing needs of users {s}<br>• Reconfigure the toolset for new satellites and relationships of satellites {t}<br>• Modular additions to displays based on user needs and financial assets (e.g., licenses) {t,c} |

s=scale, q=quality, t=time, c=cost

# Reusable Reconfigurable Scalable (RRS) Analysis

| | | | |
|---|---|---|---|
| Self-Contained Units (Modules)<br>• Table Read Modules<br>• Table Data Processors<br>• Tabular Output Modules<br>• Graphical Output Modules | Reusable | Scalable | Evolving Standards (Framework)<br>• Requirements<br>• Table Types (e.g., from different spacecraft)<br>• Enhancements |
| Plug Compatibility<br>• Common table data will process across different systems<br>• Reports build from selectable options | | | Unit Redundancy & Diversity<br>• Common Data Processing when possible<br>• The data processed will be tailored for constellation size |
| Facilitated Reuse<br>• Automated tests<br>• Common Output Formats<br>• Common processing design | | | Elastic Capacity<br>• Size and number of tables<br>• Retired modules removed<br>• New modules added |
| Reconfigurable | | | |
| Peer-to-Peer Interactions<br>• Historical data for process defects (performance metrics)<br>• Users can exchange input/output files | | | Distributed Control & Information<br>• Input data is part of output files to ensure proper data processed<br>• Output controlled by “need to know” level of user (i.e., partitioned) |
| Deferred Commitment<br>• Table composition<br>• Features of each output request | | | Self Organization<br>• Processing is determined by input and options selected |

# Customer Contact is Key

- What does the customer need?
  - There's a difference between want and need
  - A customer may comprise many diverse users
  - Show results/designs frequently enough to respond to change

- Invite customer to everything – we included customer in:
  - Design Reviews
  - Display Concepts
  - Pre-release Testing
  - Customer Surveys
  - Invitation doesn't mean they'll show up, but makes them feel welcome

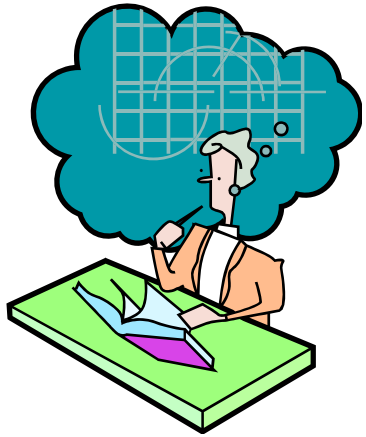
**Close customer contact = no surprises and less risk**

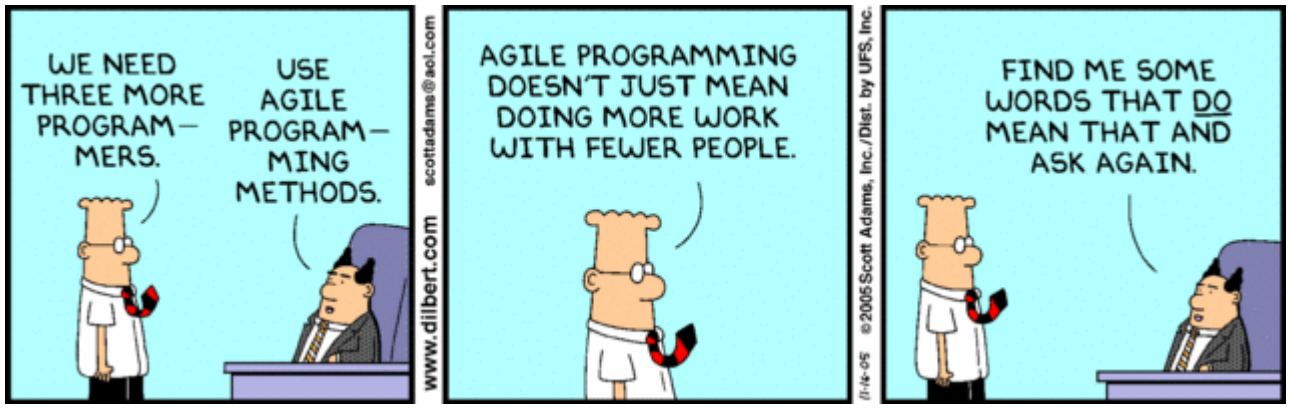
# Decision Making Aids

- Close enough for government work?
  - CEO of Best Buy and Gift Cards
  - Best Buy Store Opening
- One correct choice, is it right?
  - “Who Wants to be a Millionaire” – Wisdom of the Crowds
- Given multiple choices, which is best?
  - How Honeybees choose nest sites

**Diversity of Knowledge can help make better decisions**

- **Conclusions**








Agile Methodology instills confidence early on (risk, operational effectiveness)

Agile embraces change – indeed, change is expected/encouraged

Traditional Methodology still best for Earned Value

New methodologies necessitate new metrics

# Cost Comparisons

- Project 1 (Traditional) – 288.94 per SLOC
- Project 2 (Traditional) – 314.47 per SLOC
- Project 3 (Traditional) – 201.67 per SLOC
- Project A (Agile) – 45.71 per SLOC
- Project B (Agile) – 51.46 per SLOC

# Conclusions (Works for Agile and other methods)

NORTHROP GRUMMAN

- Close interaction and observation with customer creates opportunity
- Good relationship with customer promotes amiable environment
- Proving added value to customer results in requests for additional work
- Customer involvement key aspect of and ideal for Agile environment
- Agile environment creates useful products in short time
- Successful products with close customer/developer relationship enhances confidence, builds trust, establishes reputation
- Fits with NG Corporate theme of Performance Culture

**Happy Customers = More Work**


NORTHROP GRUMMAN