BOEING

Engineering, Operations & Technology
Boeing Test & Evaluation

Methods for Test & Evaluation Cost Efficiencies

Dennis Truong
Boeing Test & Evaluation

2013 International Test & Evaluation Association Symposium
November 13, 2013

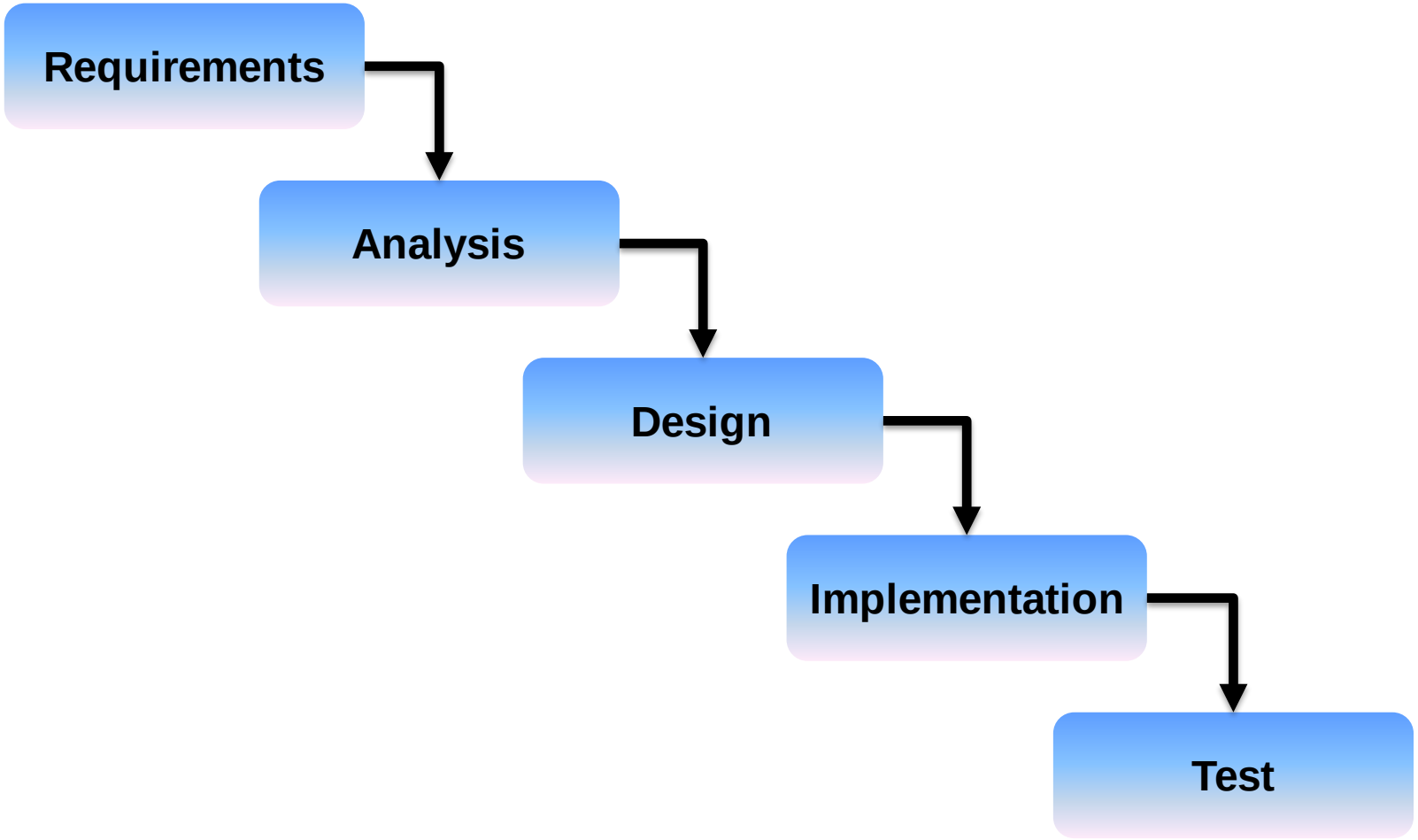
Requirements Illustration

- **“System Requirements” for a resort villa**
  - Kitchen sink shall be white
  - Kitchen sink shall have a fancy gooseneck faucet
  - Kitchen shall have windows
  - Kitchen windows shall have plantation shutters

Requirements Illustration

- **Test results:**
  - Faucet works great
  - Left shutter works great
  - Right shutter opens about an inch
- **Requirements:** met **?**
- **Impact:**
  - Alignment is assymetrical
  - Installed like this through all the villa units

Waterfall Approach



Traditional T&E

- **Independent verification**
  - Product implementation already completed
  - Product functional test and requirements verification occurs at the same time, at the end of the development lifecycle
  - Test infrastructure built separately from product development. Success depends on independent test and develop teams having congruent interpretation of System requirements
- **Problems found at this stage are costly to correct**
  - Industry-wide, rework can amount to 30% - 50% of total development cost, depending on where in the engineering lifecycle the error is caught
  - Bad requirements account for about 75% of all rework

Current Business Environment

- **“Incentivization”**
  - Financial challenges to do things faster and cheaper without quality sacrifices
  - Same amount of engineering still remains
- **Funding allocation favors development**
  - Test infrastructure considered a necessary evil
  - Customers want to focus on fielding new capabilities
- **“First Time Quality”**
  - Planning is based on optimal outcomes with very little slack for errors or pop-up problems
  - First time quality largely a myth

Cost Efficiencies

- **Using these three methods in concert yields results**
  - Reduction in deliverables
  - Upfront T&E
  - Agile development
- **Alone, each method has its own incremental gains. Combined, the gain is much greater than the sum of its parts**

Reduction in Deliverables

- **How many deliverables are truly necessary?**
- **Produce only value-added artifacts**
  - Concept is obvious but implementation is elusive
  - Often starts with customer question “how do you know…”
- **Deliverable volume often self-inflicted by defense contractor**
  - Inexperienced Program Managers agree to extraneous items in Statement Of Requirements
  - Internal processes and informal coordination documentation somehow become deliverables
  - Non-deliverables become deliverables when a precedent is set
- **Reduce or simplify tangential deliverables**
  - Multitude of EVM reports
  - Internal artifacts

Upfront T&E

- **Early T&E involvement reduces potential re-work on backend**
  - Involvement with Business Development, Customers, end-users, other stakeholders
  - Determine system and test feasibility
- **At proposal stage should have a test plan**
  - Agile roadmaps defined and properly aligned
  - Estimating based on roadmap
- **Requirements & Design**
  - Help Systems Engineering write requirements that are realistic and necessary
  - Ensure product design can be sufficiently tested with minimal test infrastructure development

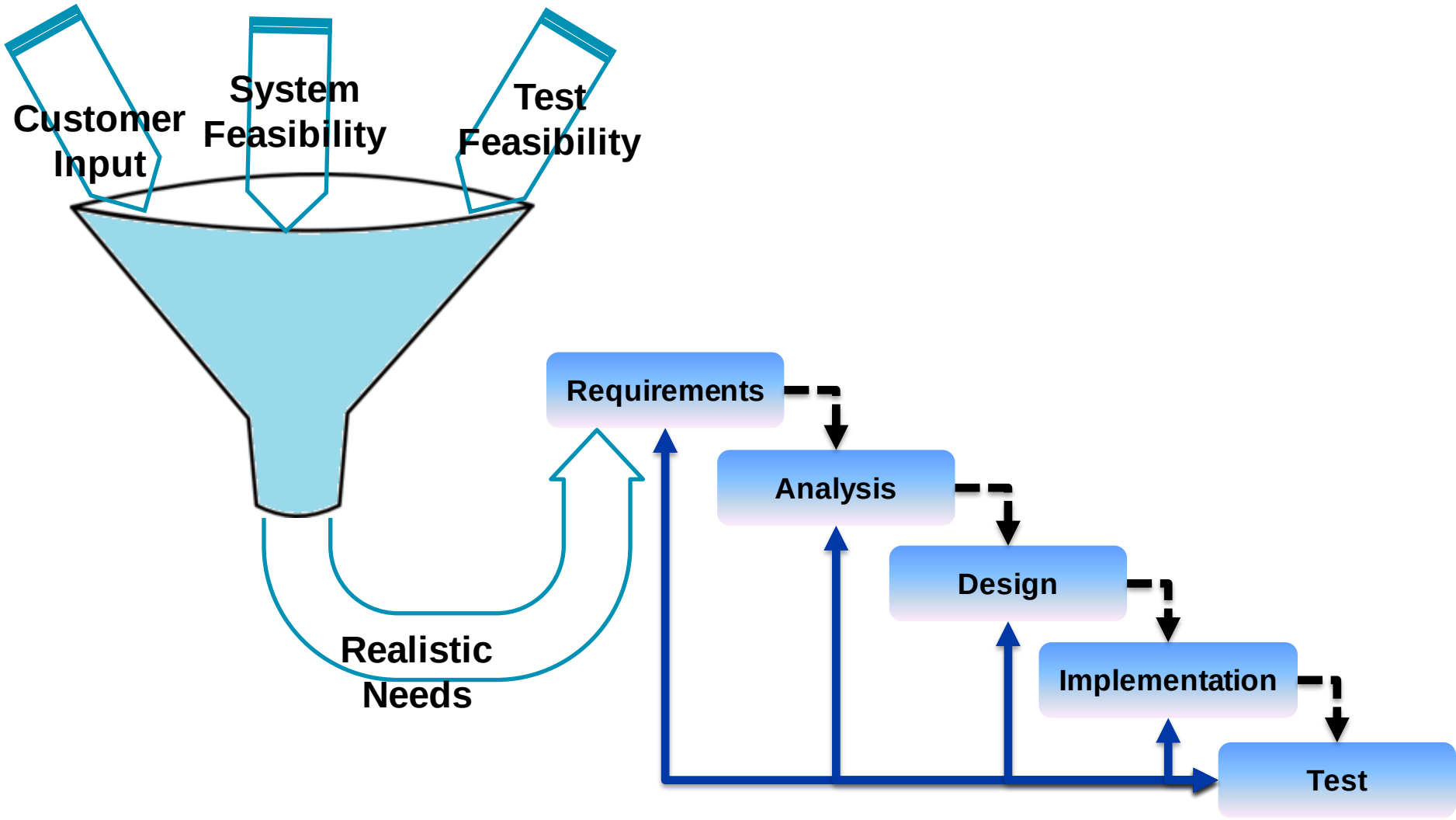
Agile

- **Agile method can be used to develop T&E infrastructure**
- **Agile decreases cost of documentation**
  - Small quantities of requirements released at a time
  - Backing documentation into completed design
    - No rework on lagging documents
    - No “peer review” paralysis
- **Agile iterations reduce the impact of rework**
  - Errors mostly confined to a Release
  - Release demonstrations immediately highlights things that are not working
  - Incremental testing
  - In the kitchen illustration, the error would have been confined to one villa unit

Agile Pitfalls

- **Agile increases cost in some areas**
  - Daily scrums and scrum of scrums
  - Release demos and iterative regressions
  - Scope creep – each scrum/iteration adds a little more “perfection”
- **Typical causes**
  - Inexperienced product owners
  - Chickens are difficult to ignore (especially when they are your management)
- **Not all customers are willing to accept Agile method**

Blended Approach



Author Biography

Dennis Truong is the Manager of Test Laboratory Systems Engineering and Software Engineering for the C-17 Program in Long Beach, CA.

Mr. Truong possesses over 13 years experience in systems engineering and software engineering. He has a proven project track record of 100% on-schedule delivery, all of which completed under budget by up to 20%. Mr. Truong leverages high performance team development in conjunction with decentralized collaboration to achieve project objectives.

dennis.truong@boeing.com

714-356-2864

Open Discussion