BOEING

INCOSE International Symposium
Philadelphia, PA
June 24-27, 2013

Can Systems Engineering be Agile? Development Lifecycles for Systems, Hardware, and Software

Dr. Ron Carson, INCOSE Fellow
The Boeing Company

Outline

- Agile principles
- Application suitability
- SE as a gated process
- Agile software engineering as continuous validation
- Key differences
- When is agile suitable for SE?

Agile Principles

- Emphasis in agile is on quickly getting to the end state vs. the process
- Application domain is Software
- Can this be extended to SE?

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.

Manifesto for Agile Software Development

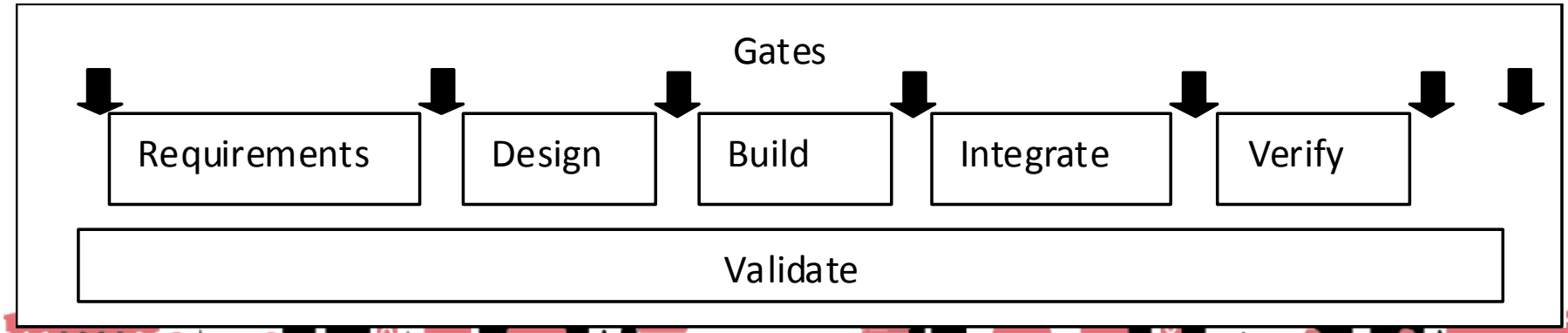
Applicability of Methods

Suitability Factors	Agile	"Plan-driven" (Traditional SE)	Formal Methods (Ross 2005)
Criticality	Low criticality	High criticality	Extreme criticality
Personnel	Senior developers	Junior developers	Senior developers
Dynamism	Requirements change often	Requirements do not change often	Requirements do not change often
Size	Small number of developers	Large number of developers	Small number of requirements that can be modeled
Culture	Culture that thrives on chaos	Culture that demands order	Rigorous modeling and extreme quality

(Ref., Boehm and Turner, 2004)

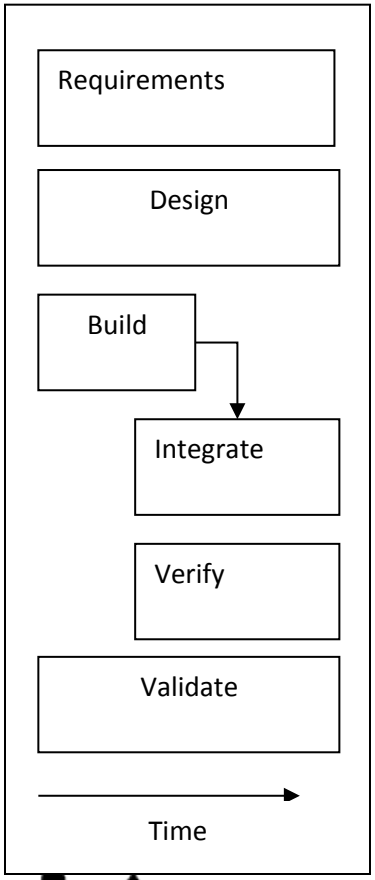
SE as an Integrated, Gated Process

- Per our standards (ISO/IEC 15288:2008, INCOSE SE Handbook), SE is a *gated* process
- The gates are regular, vertical and horizontal reviews to ensure holistic design *before Build* and that
  - All stakeholder requirements are addressed
  - All elements work together to satisfy stakeholder requirements
- "Discovery" is primarily allocated to Requirements phase



Agile SW Development as Continuous Validation

- Agile emphasizes following the user – continuous validation
- "Working software" may be releasable – no need for additional "manufacturing"
- Designer may be the builder
- SW development can adopt "finish-to-finish" strategy
  - Build it first
  - Confirm user satisfaction
  - Subsequent builds assumed to cause little rework
  - Document and Verify as needed
- Enables rapid change as needed



Key Differences

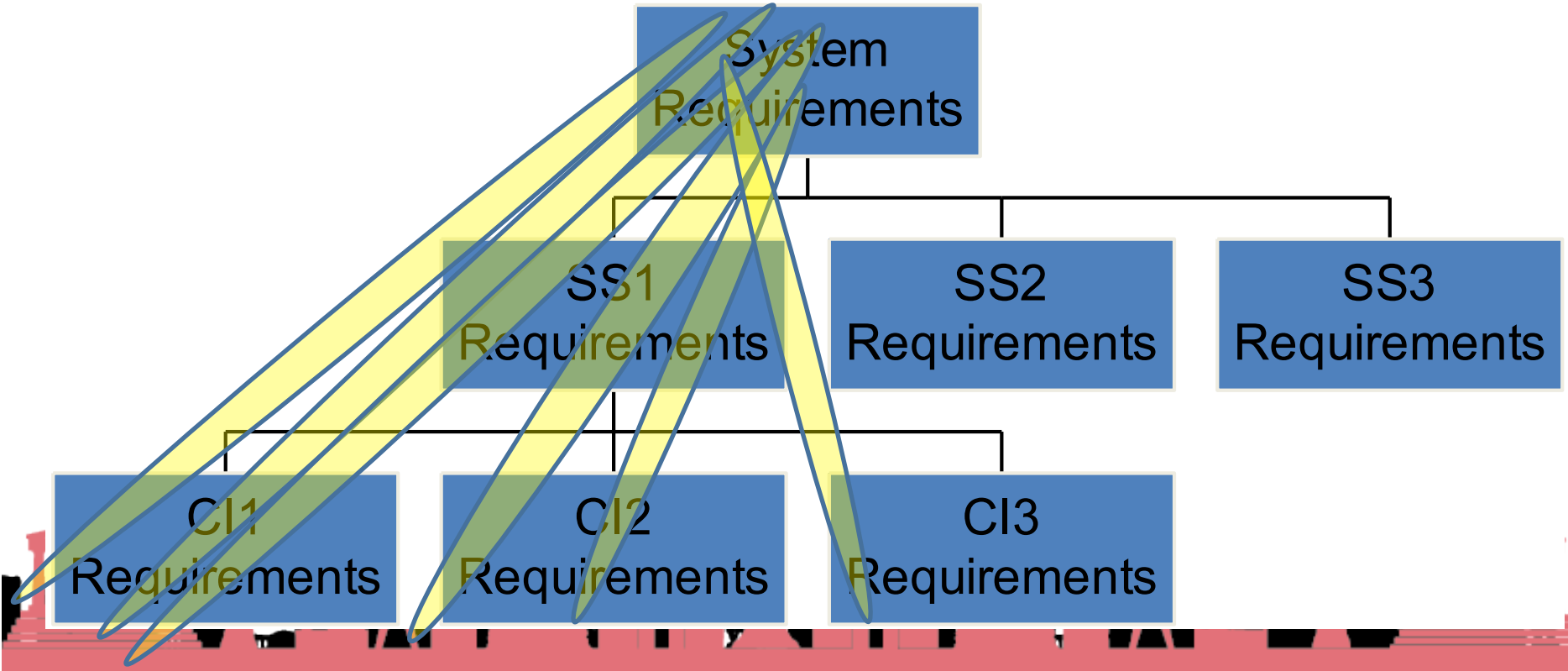
Topic	SE	Agile SWE
Discovering what users value	Up front; formal, incremental update	Continuous discovery and refinement

When is Agile Suitable?

Suitability Factors	Agile	Plan-driven (Traditional SE)	Formal Methods (cf., Ross 2005)
Program application	Internal or exploratory development	Contracted development	Contracted development

"Agile SE" Process

- "Agile SE" would be characterized by an ability to create "vertical slices" of capability, incrementally
  - Vertically integrated; may not be horizontally integrated
  - Continuous discovery may lead to "architecture breakers"



Enablers for Agile SE

- Contracting by other than a specification
- Decrease subsystem coupling
- Reduce hardware/system fabrication cycle time
  - 3-D print/fabrication
  - Continuous integration of changes
- Stable infrastructure for things that take a long time to change

Summary

- Unique features of typical SW development enable benefits of agility
  - Continuous discovery
  - Rapid update
- "Agile SE" carries additional development risk
  - Risk in horizontal integration: "architecture breakers"
  - Risk of volatility due to additional user involvement
  - Risk in not seeing the whole, integrated design
  - Overall risk of additional rework