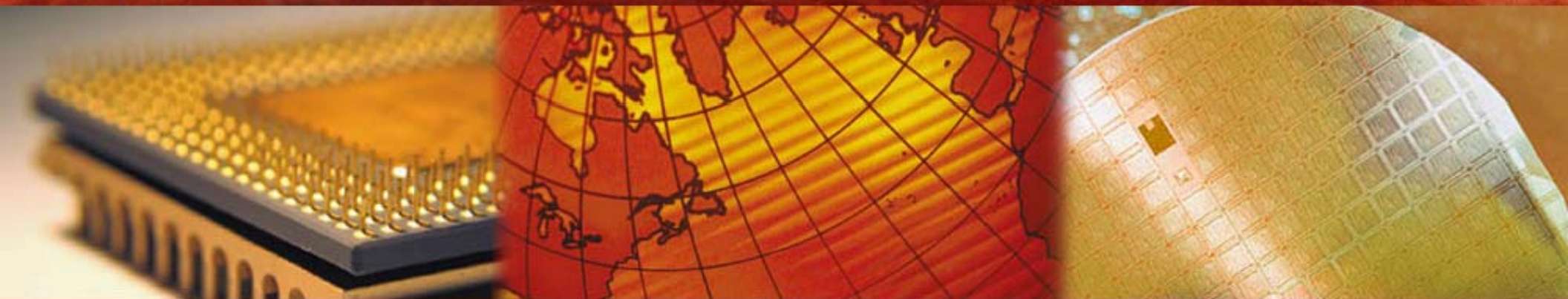




Using Agile Techniques with ESL for Better Results

David Black, Hemendra Talesara, Neil Johnson



Agenda

- Agile context
- ESL context
- Closer look at Agile
- From Waterfall to Agile
- How Agile benefits ESL

www.agilemanifesto.org

Defined Approach
(Very Rigid)

Agile Approach
(Highly Adaptive)

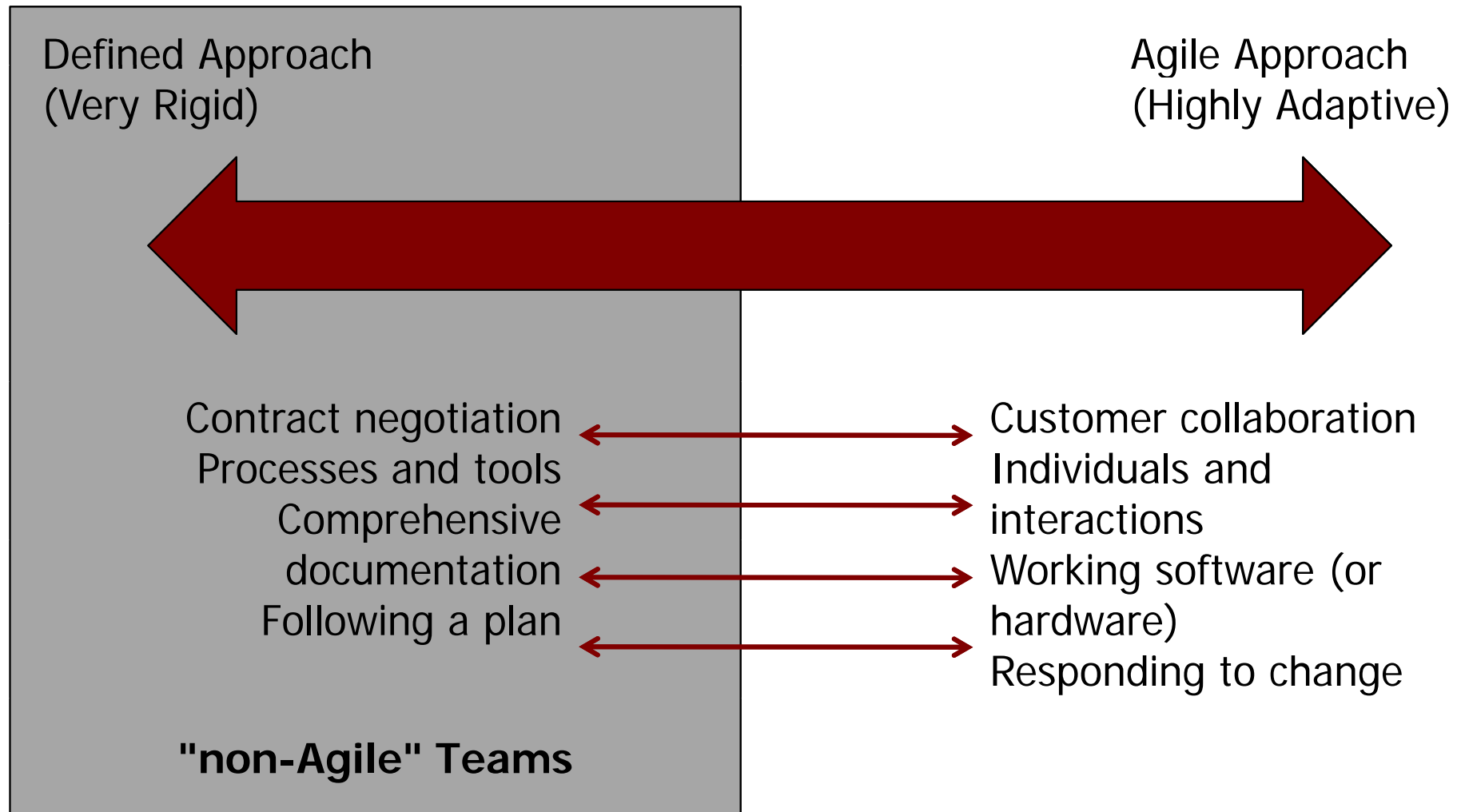


Contract negotiation
Processes and tools
Comprehensive
documentation
Following a plan



Customer collaboration
Individuals and
interactions
Working software (or
hardware)
Responding to change

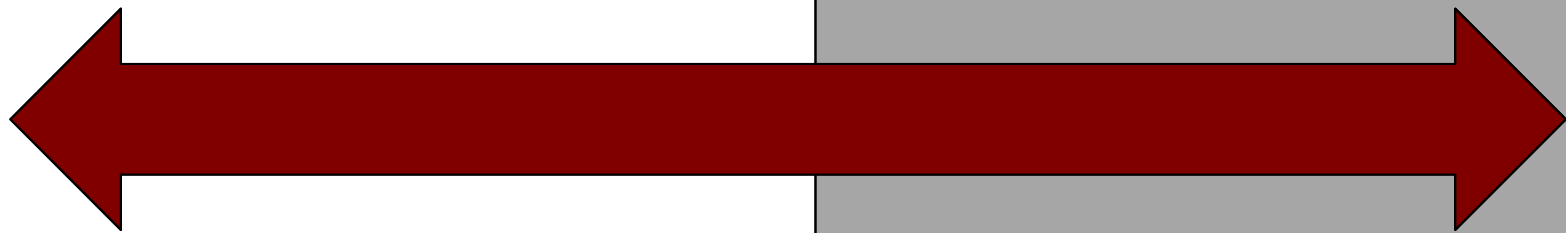
www.agilemanifesto.org



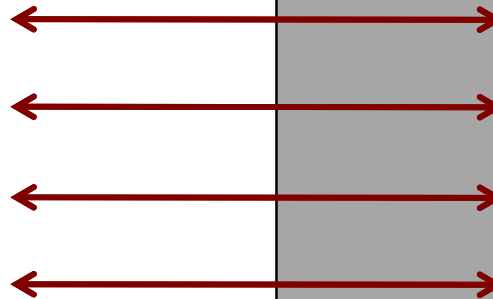
www.agilemanifesto.org

Defined Approach
(Very Rigid)

Agile Approach
(Highly Adaptive)



Contract negotiation
Processes and tools
Comprehensive
documentation
Following a plan



Customer collaboration
Individuals and
interactions
Working software (or
hardware)
Responding to change

Agile Teams

Agenda

Agile context

ESL context

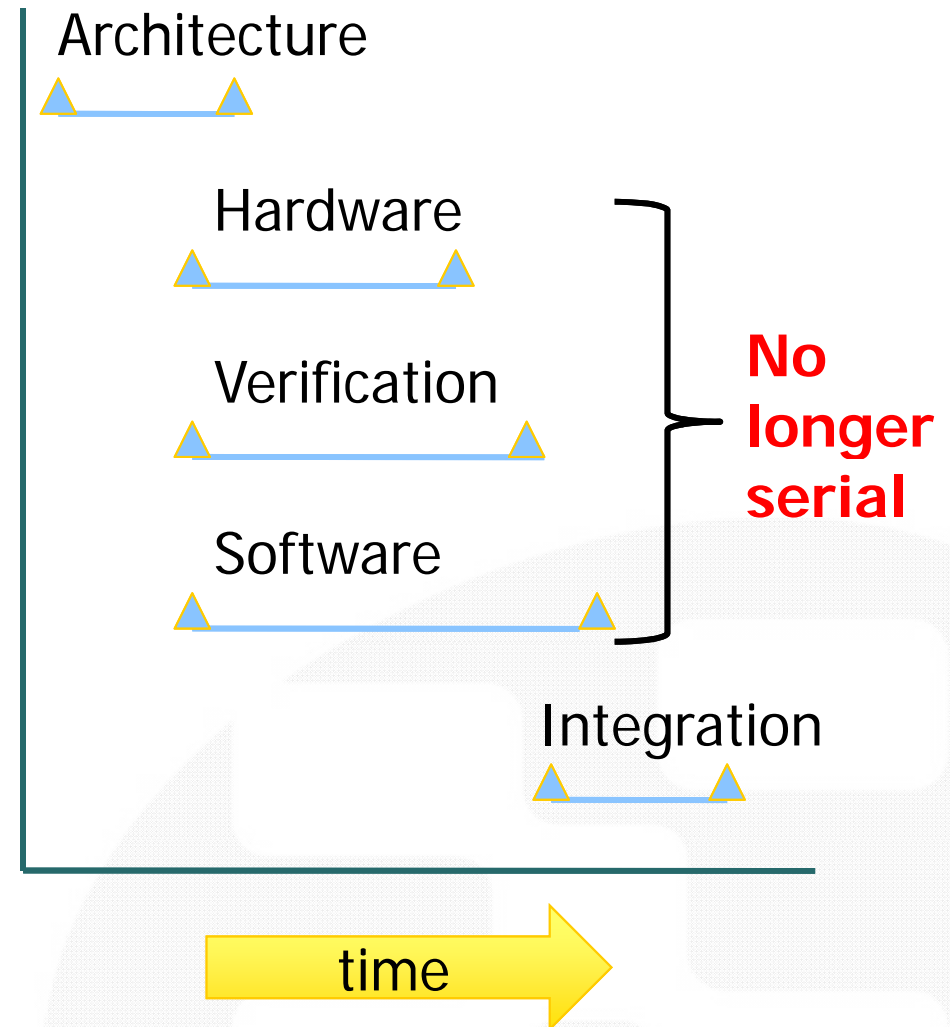
Closer look at Agile

From Waterfall to Agile

How Agile benefits ESL

ESL Improves Scope & Scheduling

- DEFINITION: Electronic **System-Level** design
 - Level of abstraction encompassing all aspects of the electronics portion of the system including hardware, software, and surrounding environment



ESL Use-Cases

Models for Architecture Exploration

Models for Performance Analysis

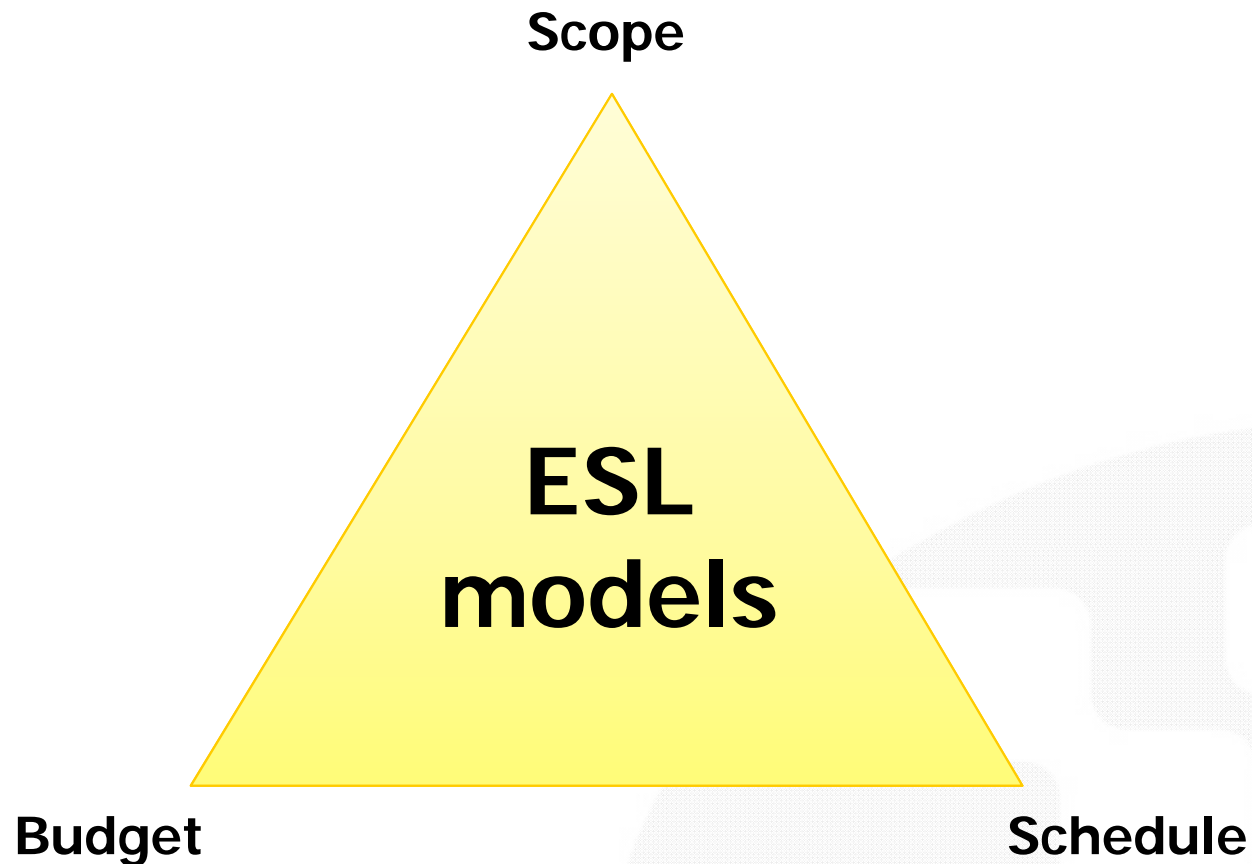
Models for Early SW Development

Models for Early Verification

Models for High Level Synthesis

ESL MODEL PROVIDES AN EXECUTABLE SPECIFICATION

ESL Goal: Understand the big picture



Agenda

Agile context

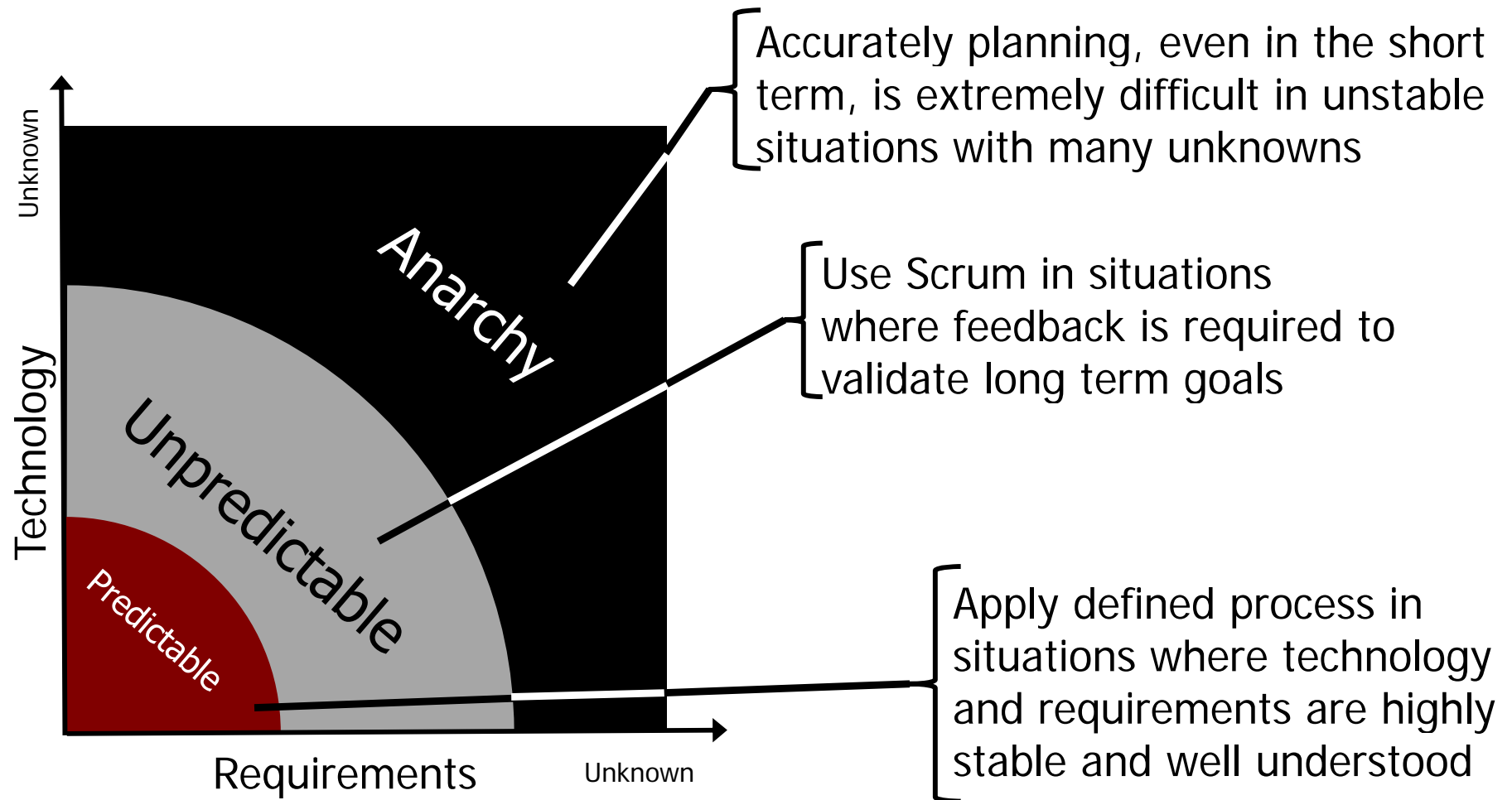
ESL context

Closer look at Agile

From Waterfall to Agile

How Agile benefits ESL

Empirical vs. Defined Process



Working Software (Hardware)

Waterfall Model

- A sequential process
- One big bang, production ready release at the end of the project
- Lessons learned for the next project
- Task driven development

Working Software (Hardware)

Waterfall Model

- A sequential process
- One big bang, production ready release at the end of the project
- Lessons learned for the next project
- Task driven development

• Agile Model

- An incremental process
- Many production ready “re-spins” during the project
- Regular retrospectives
- Deliverables driven development

Working Software (Hardware)

Waterfall Model

- A sequential process
- One big bang, production ready release at the end of the project
- Lessons learned for the next project
- Task driven development

• Agile Model

- An incremental process
- Many production ready “re-spins” during the project
- Regular retrospectives
- Deliverables driven development

Agenda

Agile context

ESL context

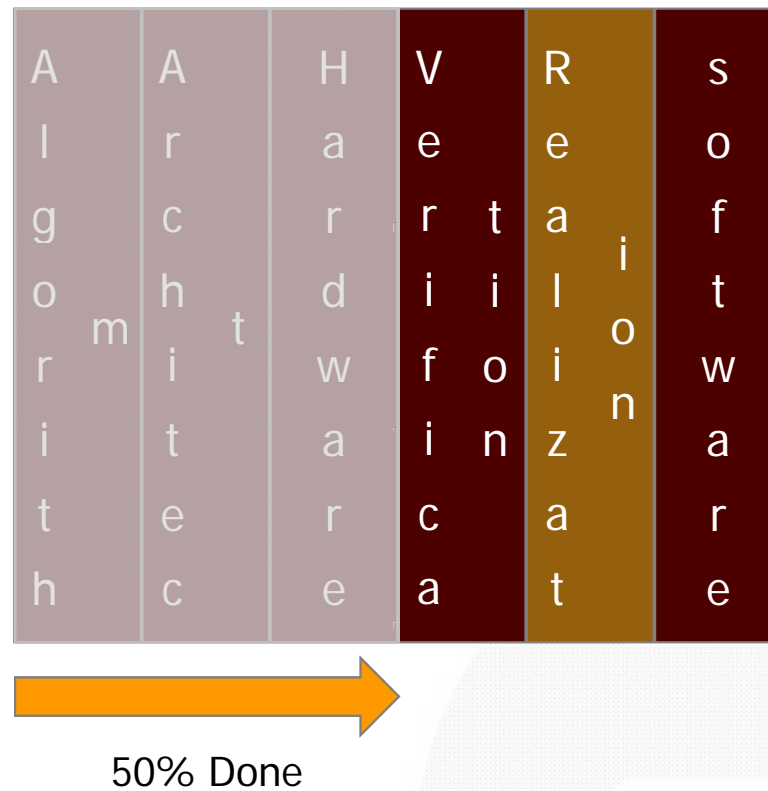
Closer look at Agile

From Waterfall to Agile

How Agile benefits ESL

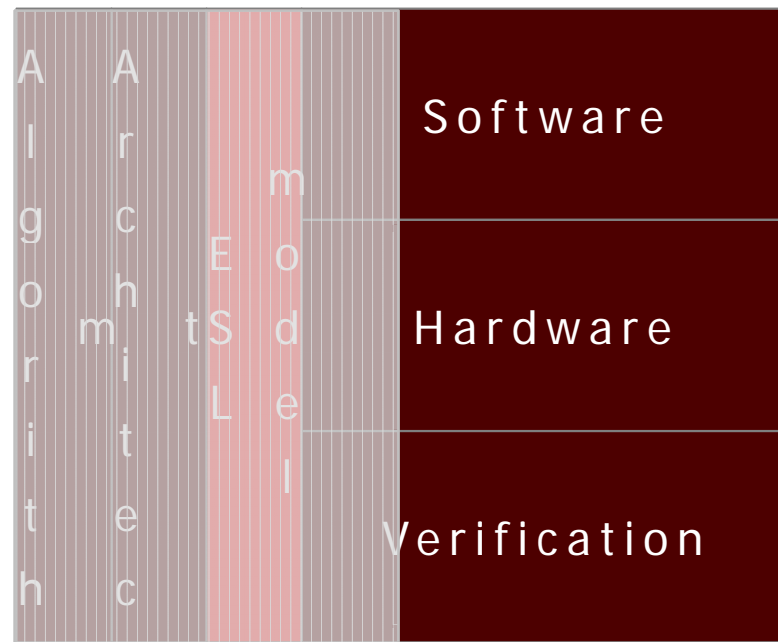
From Waterfall to Agile

Traditional waterfall (prior to ESL)



From Waterfall to Agile

ESL model enables concurrent start of implementation
 Removes Realization requirement for Software



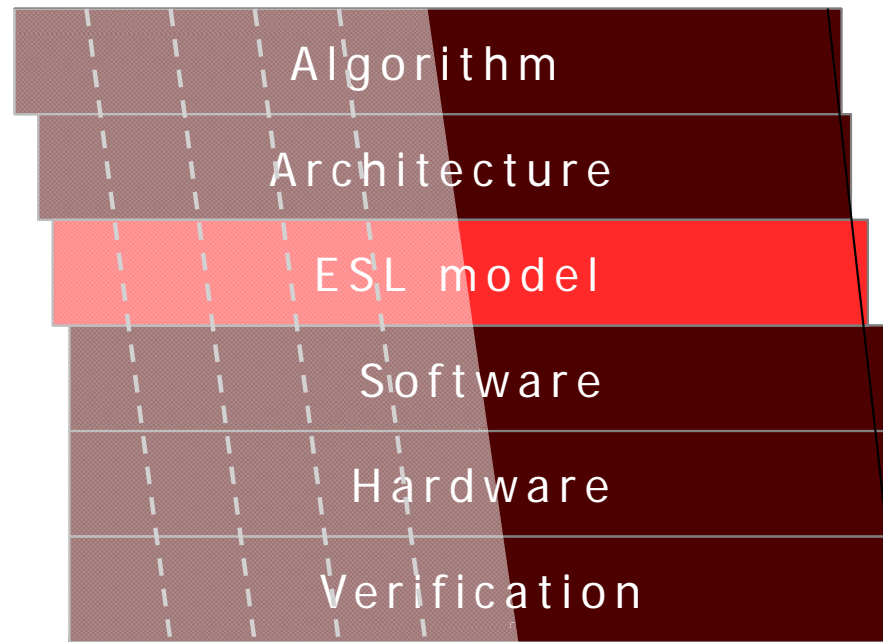
50% Done

50% Done



From Waterfall to Agile

Agile enables incremental design and quicker feedback



50% Done

Agenda

Agile context

ESL context

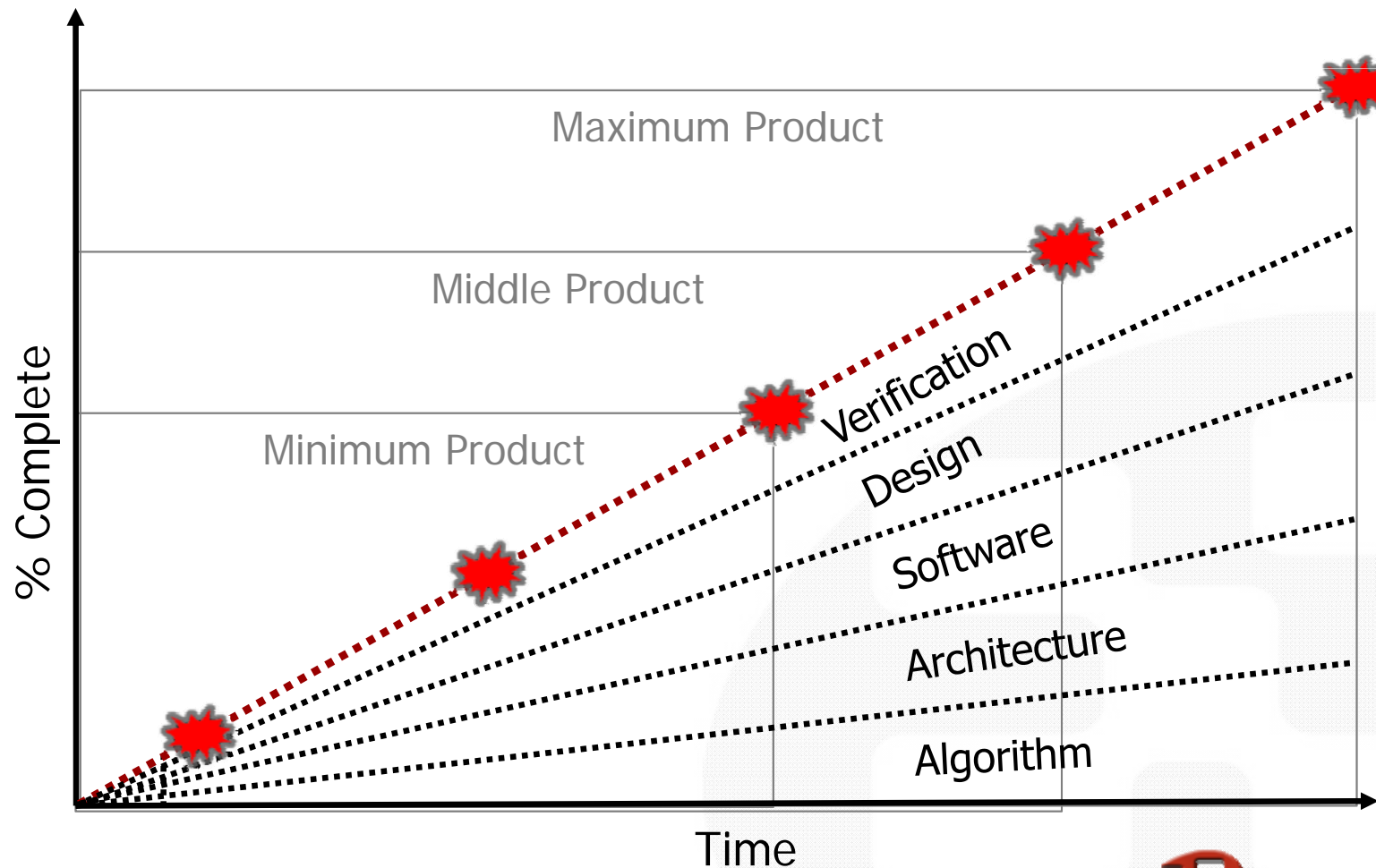
Closer look at Agile

From Waterfall to Agile

How Agile benefits ESL

Incremental Development

Agile development teams regularly provide deliverables



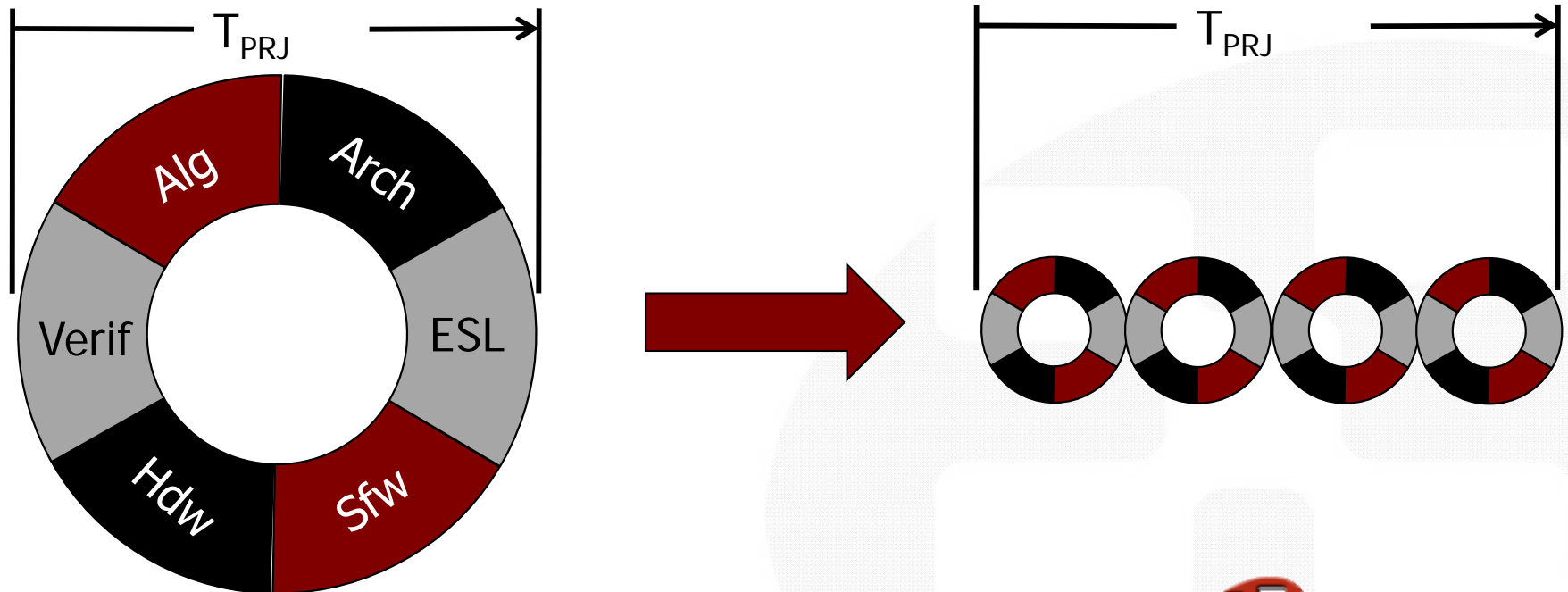
What Does Agile Mean to Us Now?

Agile planning model

- Defined process hasn't worked very well

Incremental development to replace the big bang

- Better teamwork
- Feature based milestones w/functional, demonstrable code



Agile + ESL = Natural Fit

1. Agile is successive refinement
 2. Agile is successful for software development
 3. Agile is a progression of changes/features
1. ESL authored for reuse and refinement
 2. ESL modeling "is" software
 3. ESL supports mixing levels of abstraction
 - Untimed, Loosely timed, Approximately timed, ...

Agile project tools manage processes inherent to ESL

Summary: Benefits of Going Agile

More productive team

- Visibility and effective communication
- Cross-functional development team
- Regular retrospection
- Customer collaboration

Progressive feature closure

- Responsive to change
- Incremental development
- Continuous integration

Big value to ALL stake holders - win - win - win

References

- www.synopsys.com/Community/SNUG/Sanjose/

www.agilesoc.com/articles

- *Why Agile is a Good Fit for ASIC and FPGA Development*
- *Agile Transformation in IC Development*
- *An Agile Approach to ESL Modeling*
- *Agile Teams: What Are They Good For?*
- *And others...*

Books

- *Succeeding with Agile: Software Development Using Scrum* by Mike Cohn
- *Extreme Programming Explained: Embrace Change (2nd Ed)* by Kent Beck

