When will we be able to verify real hybrid systems?

Bruce H. Krogh
Carnegie Mellon University

Overview

- From research to regular use
- Example: Model checking
- Some possible domains for HS verification
- Summary
- References

Technology Maturation Process (Redwine & Riddle)

- Basic Research
- Concept Formation
- Development and Extension
- Internal Exploration
- External Exploration
- Popularization

Technology Maturation Process (R&R)

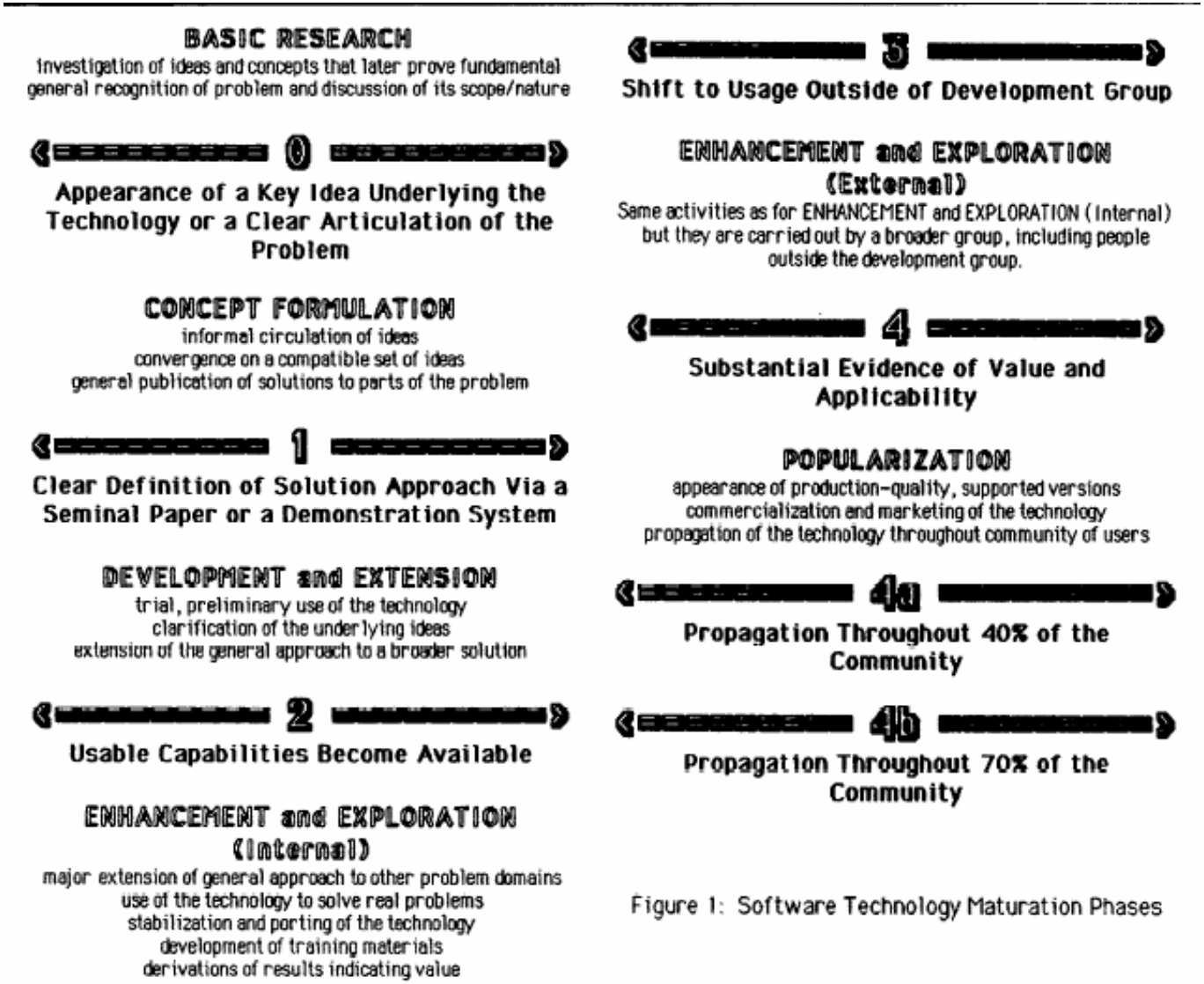


Figure 1: Software Technology Maturation Phases

Development of Model Checking (Polidian)

Basic Research ==> Concept Formation

- **1977-81: synthesis of code for data abstraction / temporal logic**

Concept formation ==> Development and Extension

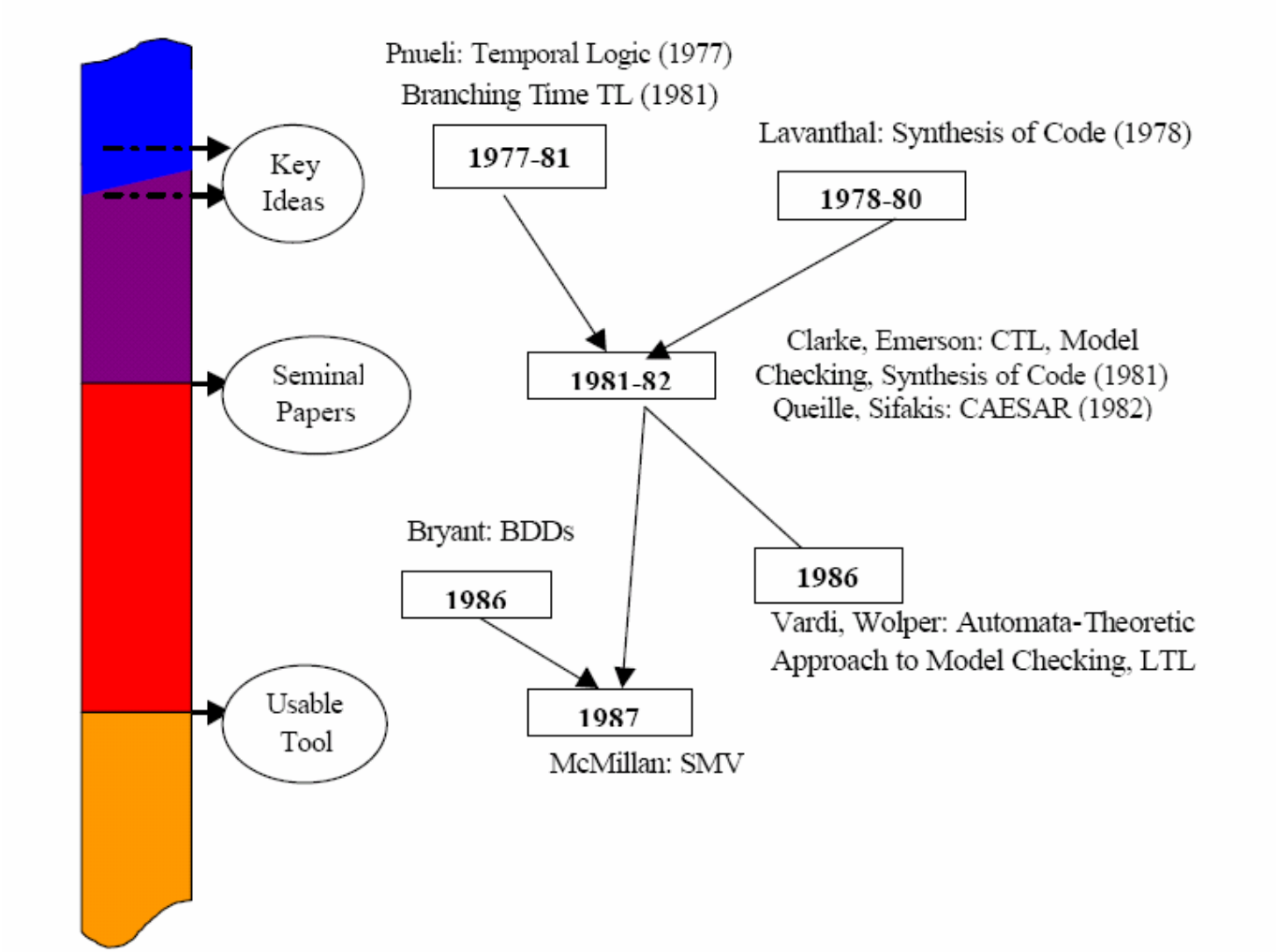
- **1981-82: model checker algorithm, seminal papers**

Development and Extension ==> Internal Exploration

- **1986-87: BDDs, SMV: a usable model checker**

- **Basic Research**
  - 60's-70's Floyd, Hoare, Dikstra, Pnueli
- **Concept Formation**
  - early 80's: Queille & Sifakis, Clarke & Emerson
- **Development and Extension**
  - late 80's: McMillan (SMV), Holzmann (SPIN)

Development of Model Checking – Early Years



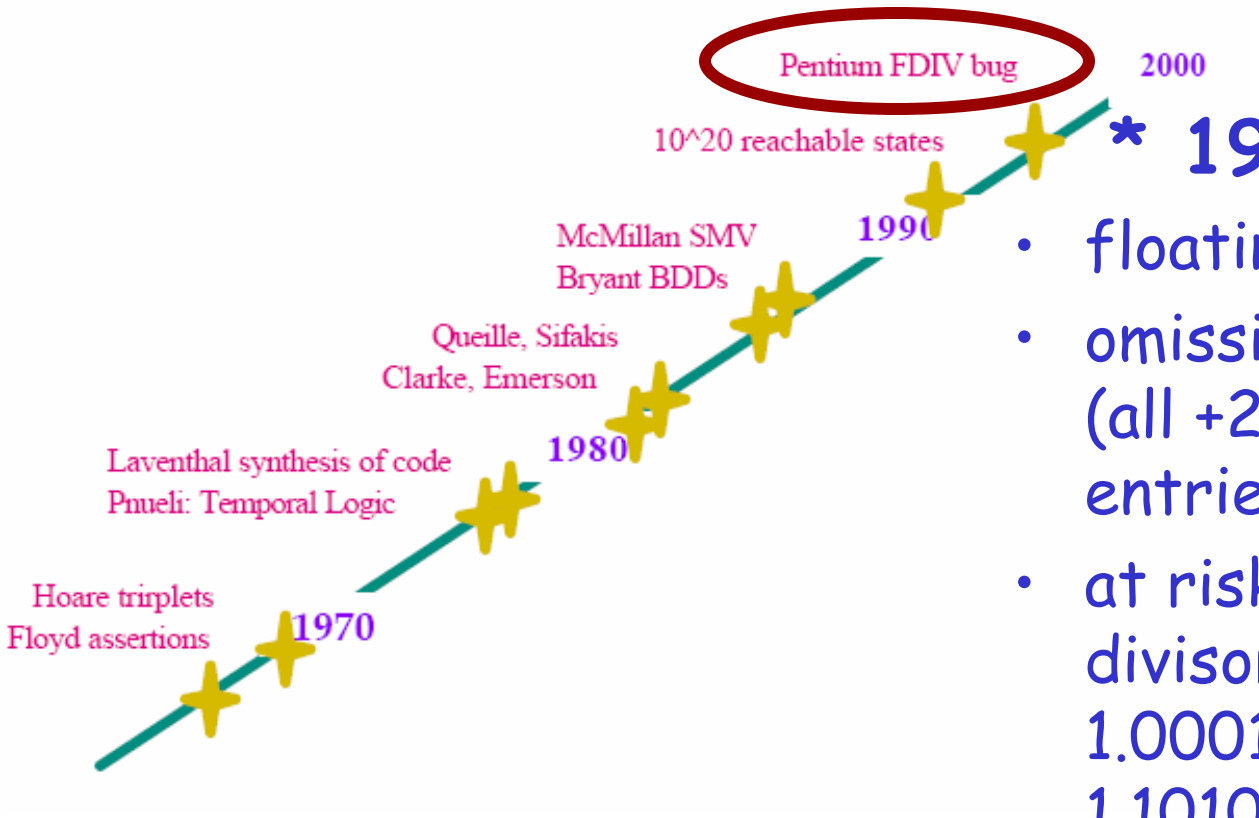
Development of Model Checking (cont'd)

- **Internal Exploration**
  - early 90s: several internal industrial projects
- **External Exploration**
  - late 90s: commercial products (Kurshan '97)
    - Abstract Hardware Ltd. (CheckOff-core technology developed at Siemens)
    - Chrysalis (Design Verifyer)
    - Compass (VFormal- core technology developed at BULL)
    - IBM (RuleBase- core technology developed at CMU),
    - Lucent Technologies (FormalCheck)

Maturation of Model Checking

- **Popularization (2000's)**
  - Standard tool in the digital design tool chain
    - E.g., Cadence
      - *Incisive - Functional Verification of RTL*
      - *Encounter Conformal – Equivalence checking (get figure)*
    - INTEL FORTE Verfication Enviornement
      - *http://www.intel.com/technology/silicon/scl/fortefl.htm*
  - Microsoft SLAM Project for driver verification
    - http://research.microsoft.com/slam/

A Key Event for Model Checking: The Pentium FDIV Bug



* 1994:

- floating pt, division error
- omission of five entries (all +2) in a table of 1,066 entries for SRT division
- at risk: divisors binary divisors beginning with 1.0001, 1.0100, 1.0111, 1.1010, and 1.1101.
- CPU recall cost Intel $500,000,000[1]

*from Ivars Peterson's **MathLand**; [1] Fix & McMillan

Role of Verification in the Digital Hardware Design Flow (Fix & McMillan)

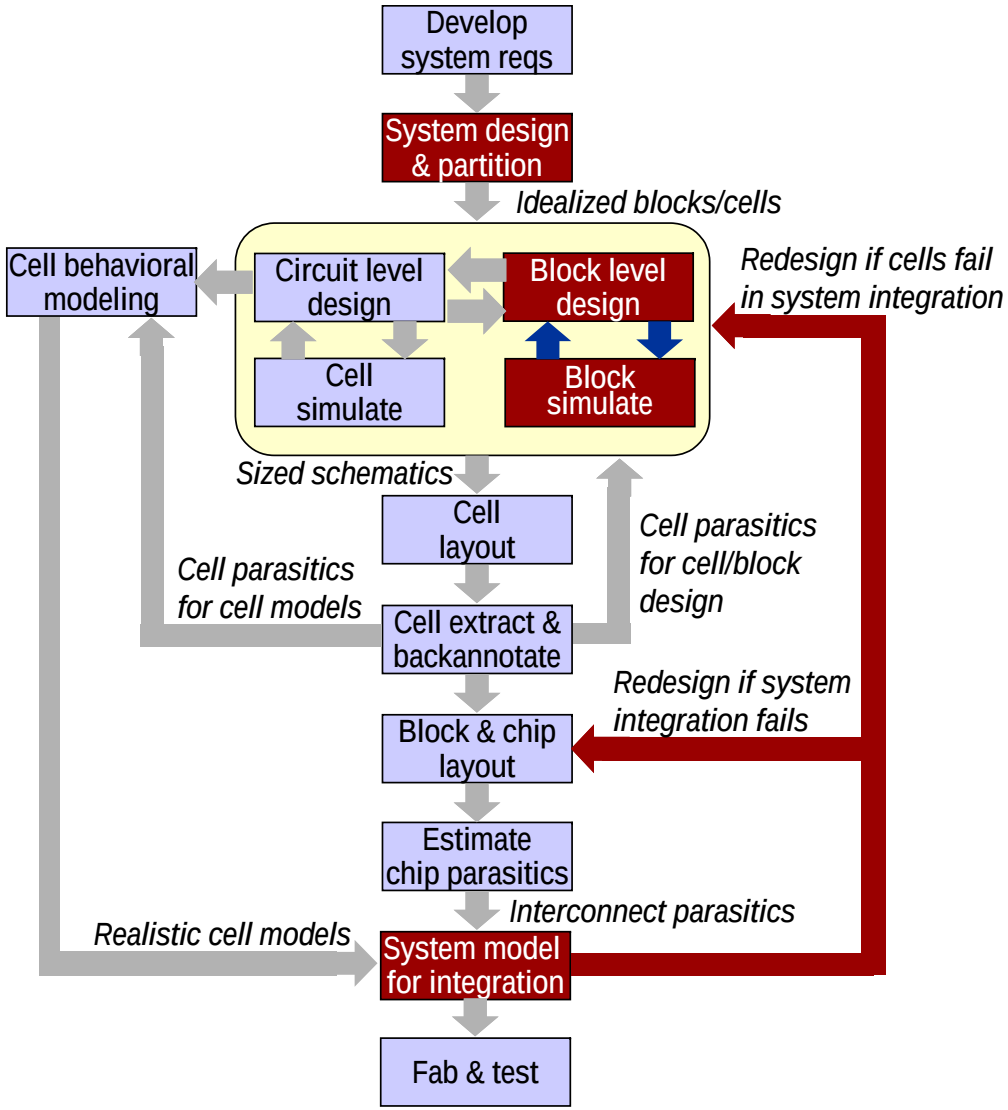
Project Phases		Design	Stabilization	Implementation	Convergence	Debug
RTL		uArch development	RTL development	RTL validation		
Timing		Timing specification at block level	Full timing specification	Converging timing to project goals	Timing converged	
Circuit		Schematic entry	25% schematic	100% schematic	Only bug fix in schematic	
Layout				Layout clean at block level	Layout assembled and clean	
Post-silicon					A0 tap-out	Functional Silicon
FPV		Verify Micro Architectural properties on FV models				
			Verify RTL properties			
				Verify Micro Architectural properties on the RTL		

Figure 1: Formal Property Verification in Hardware Design Flow

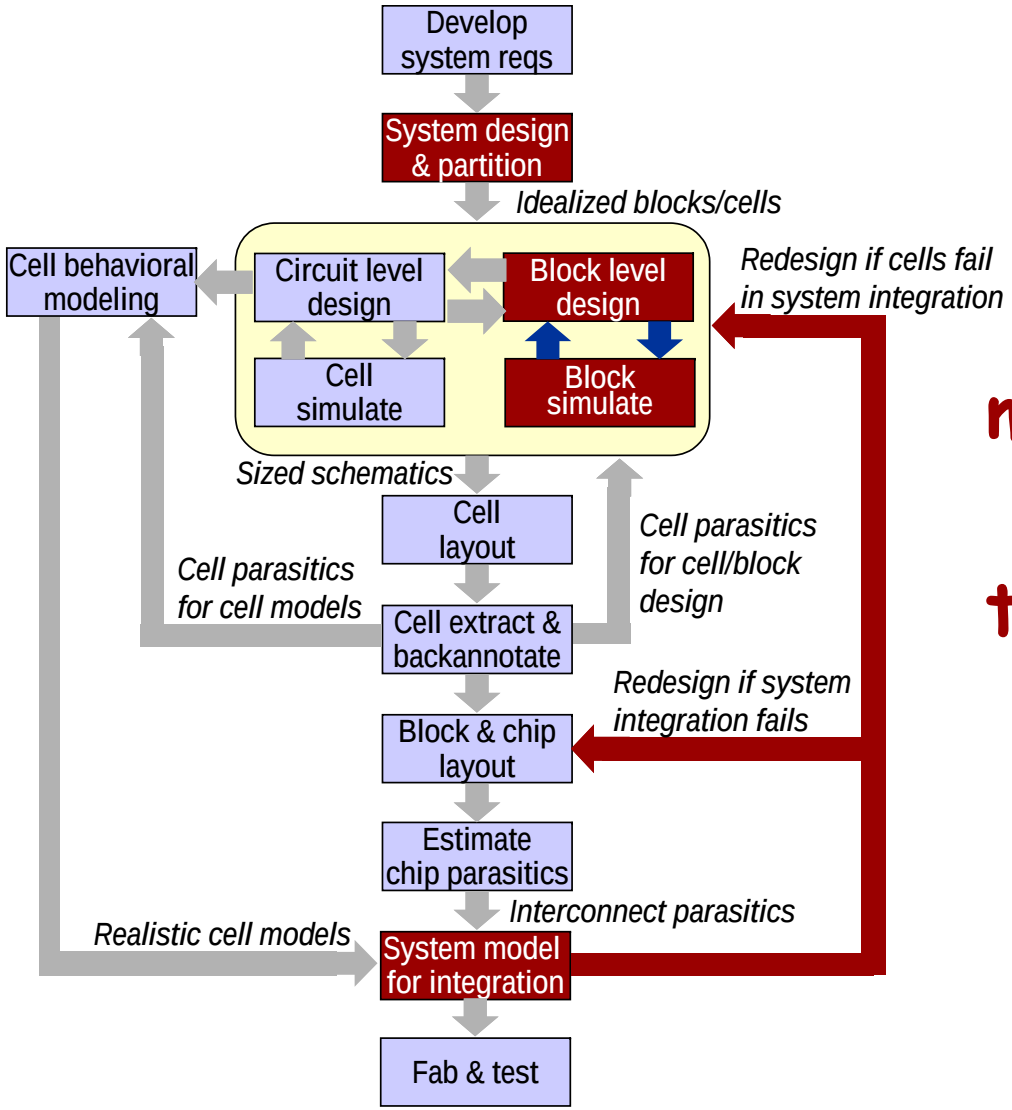
Potential Applications of Hybrid System Verification (HSV)

- **Target domains**
  - Analog Circuits
  - Flight Control Systems
  - Automotive Systems
- **Issues for each domain**
  - Where can HSV impact the design flow?
  - What's the current status of HSV for these applications?
  - Some key barriers?

Analog/Mixed (A/M) Circuits – Design flow



Analog/Mixed (A/M) Circuits – Design flow



main analysis tool:
simulation

time scales:
over lunch,
over night, and
over the weekend

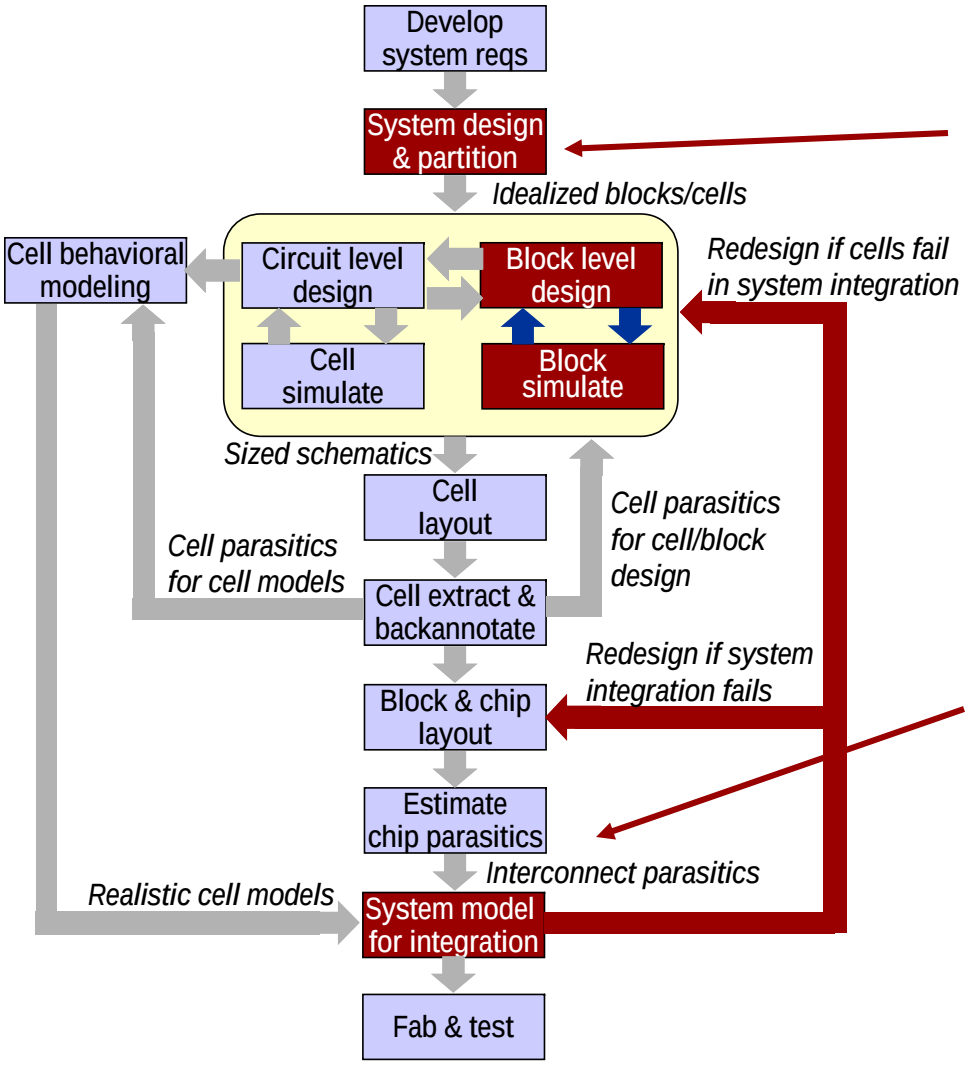
Motivation for the Verification of Analog/Mixed-Signal Circuits

- By 2006, 75% of all chips will include analog circuits.
- While these circuits only make up 2% of the devices and 20% of the area, they are taking 40% of the design effort.
- About 50% of errors requiring redesign are due to errors in the analog portions.
- Therefore, improvements in analog circuit validation methodology are becoming increasingly important.

Data on this slide is from IBS Corp's industry reports (2003).

(slide from Chris Meyers)

A/M Circuits – Possible applications of HSV



Initial verification problem

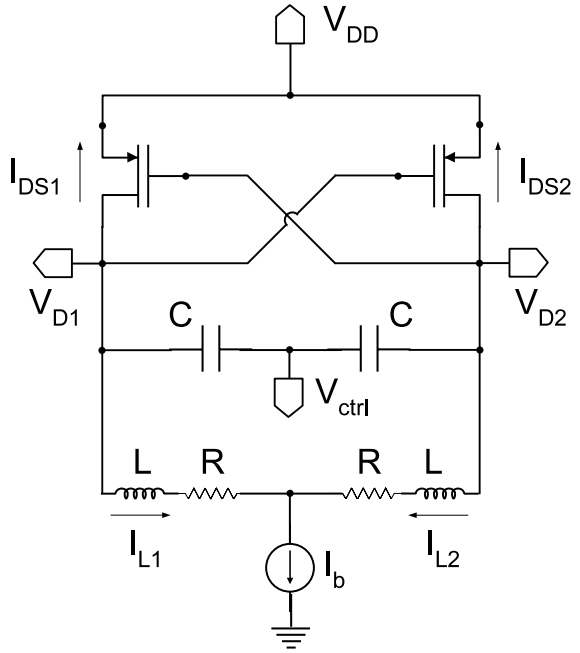
- Check **early** if there are problems with the spec or with the idealized initial design

System integration verif. problem

- Check **late** for problems caused when ideal blocks become real circuits with unwanted but unavoidable behaviors

A/M Circuits – Current status of HSV

- Academic demonstrations for small circuits
- Handful of conference papers
- 2005 Workshop on Formal Verification of Analog Circuits (a satelite event of ETAPS 2005)

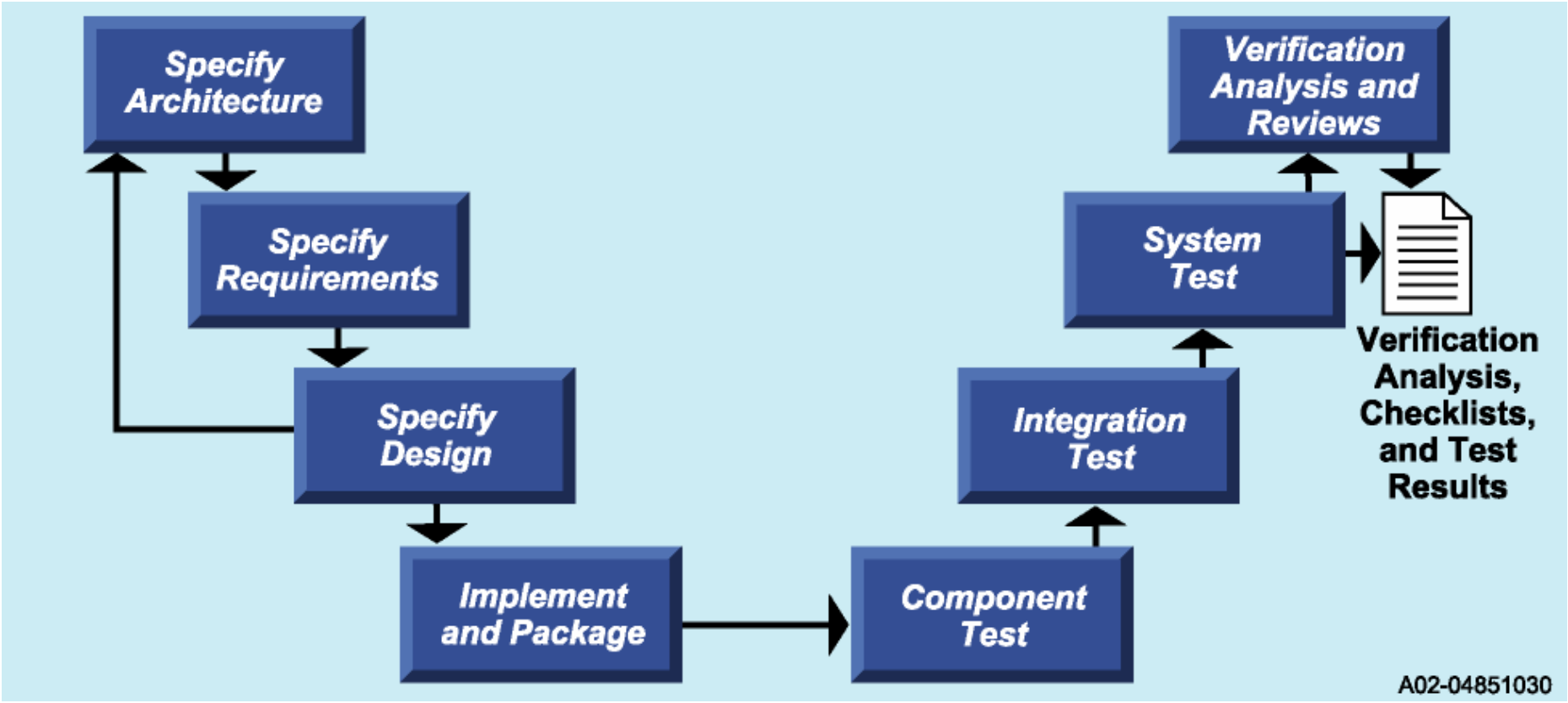


VCO (Frehse et al.)

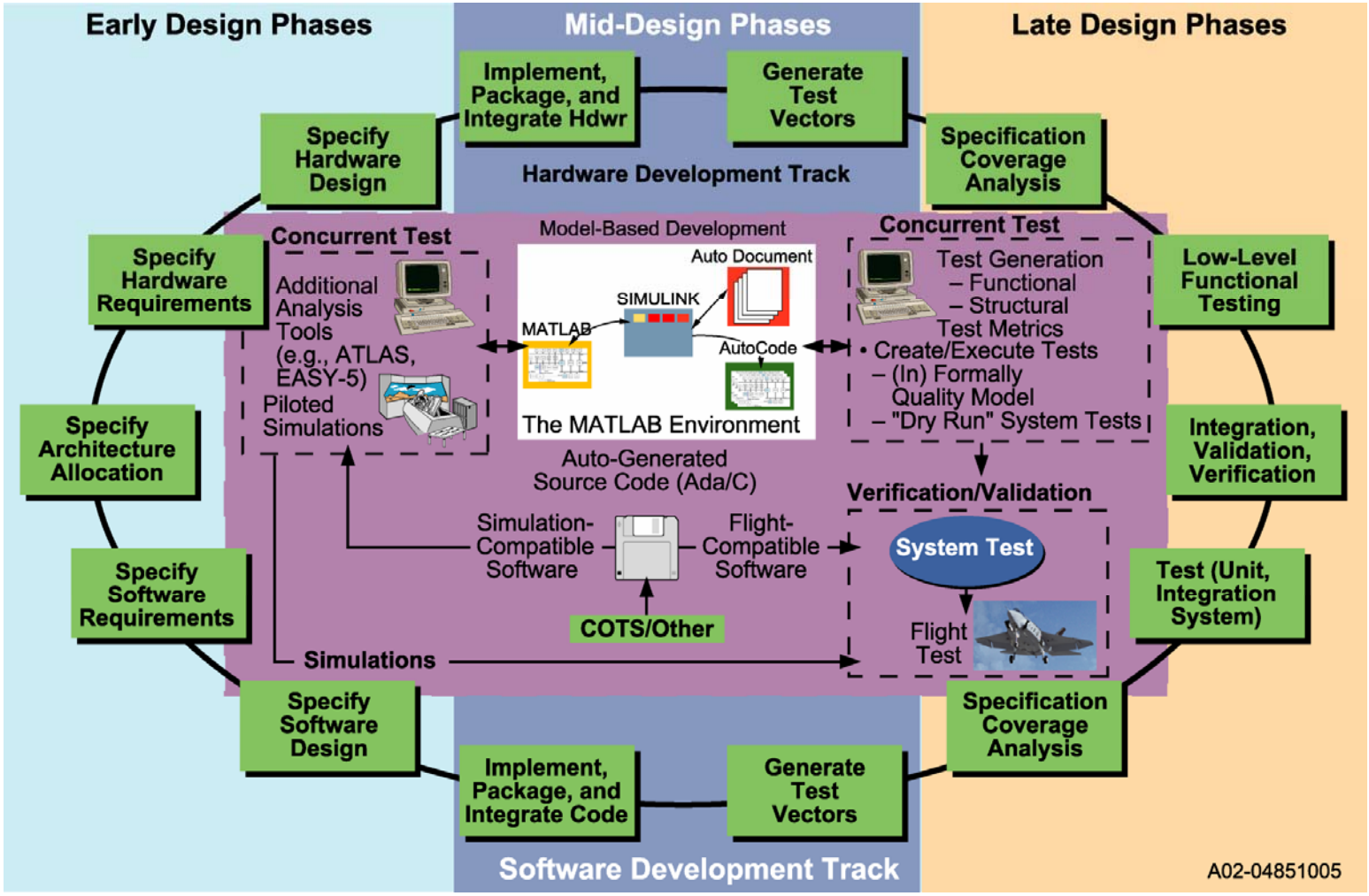
A/M Circuits – Barriers for HSV

- Expert designers consider it an art
- Formal methods are not used
- Models are not typically compatible w/ HSV tools
  - circuits described as transistor netlists
- Even SPICE is not fully trusted
- Specifications are typically in the frequency domain

Flight Control Systems (FCS) – Design flow (Buffington et al.)



FCS – Design flow (cont'd)

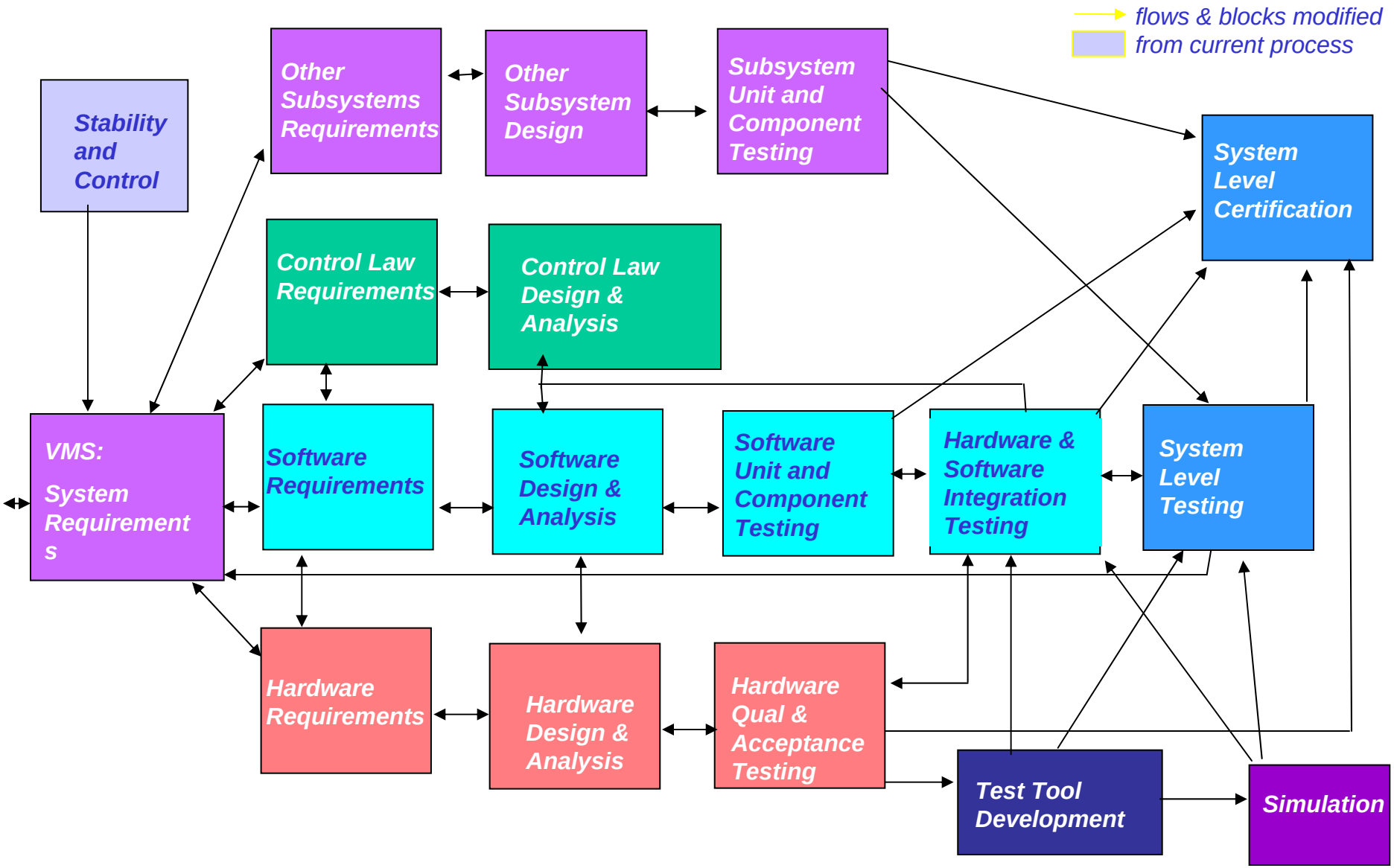


Model-based Environments
Formal Specification Techniques
Advanced V&V-aware Designs

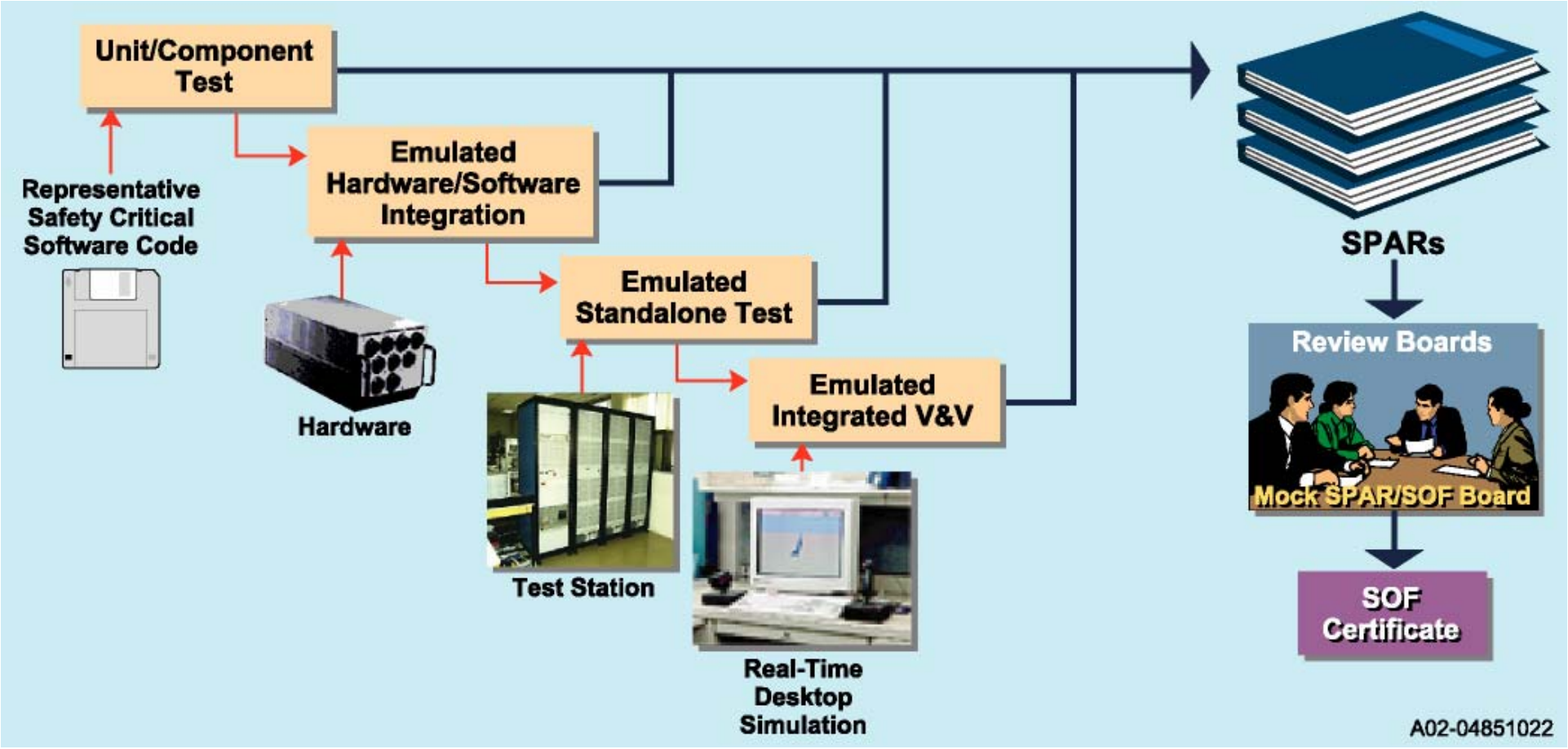
Control Analysis
Software Implementation
Formal V&V

Automated Test
Process-Based Certification

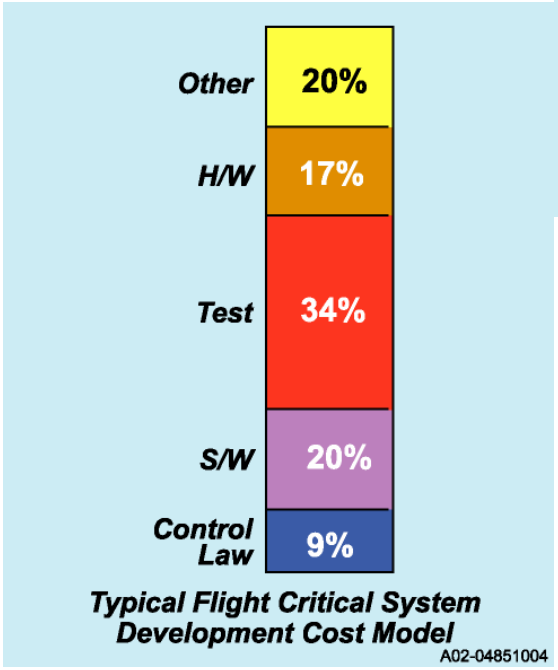
Current Process

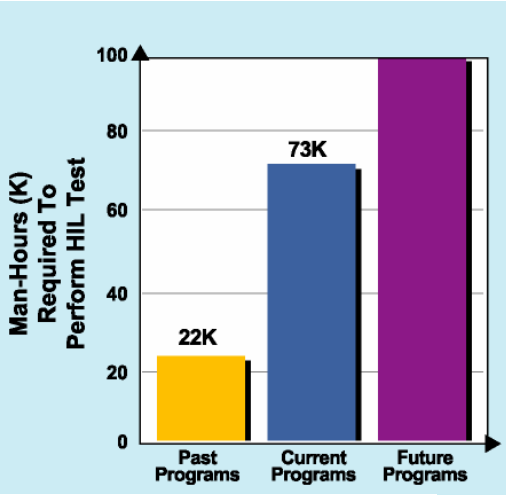


FCS – Verification via Testing

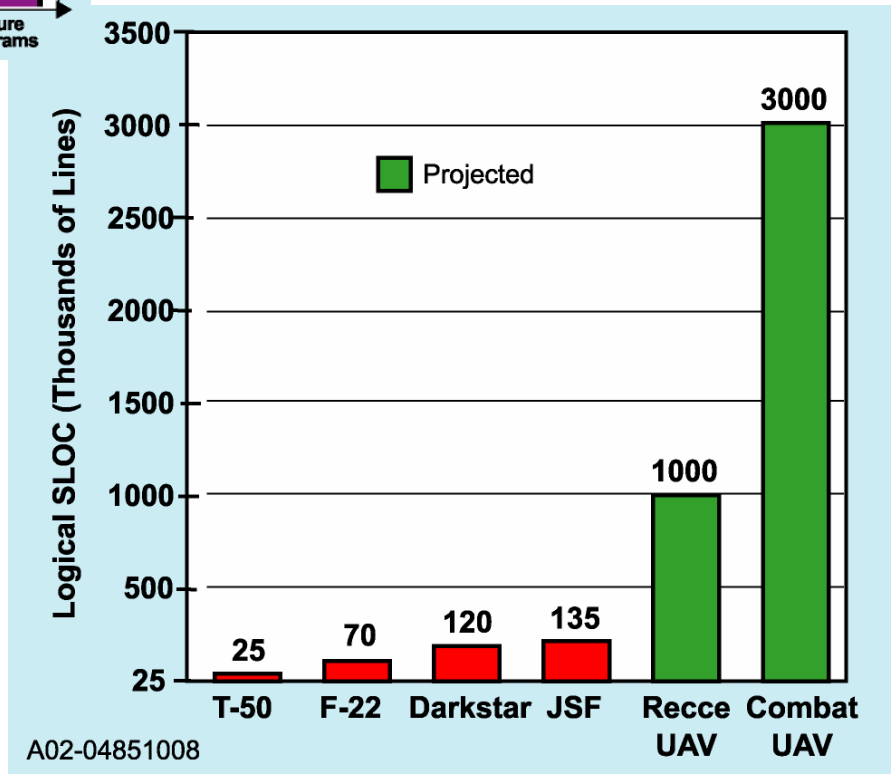


FCS – Cost of Testing

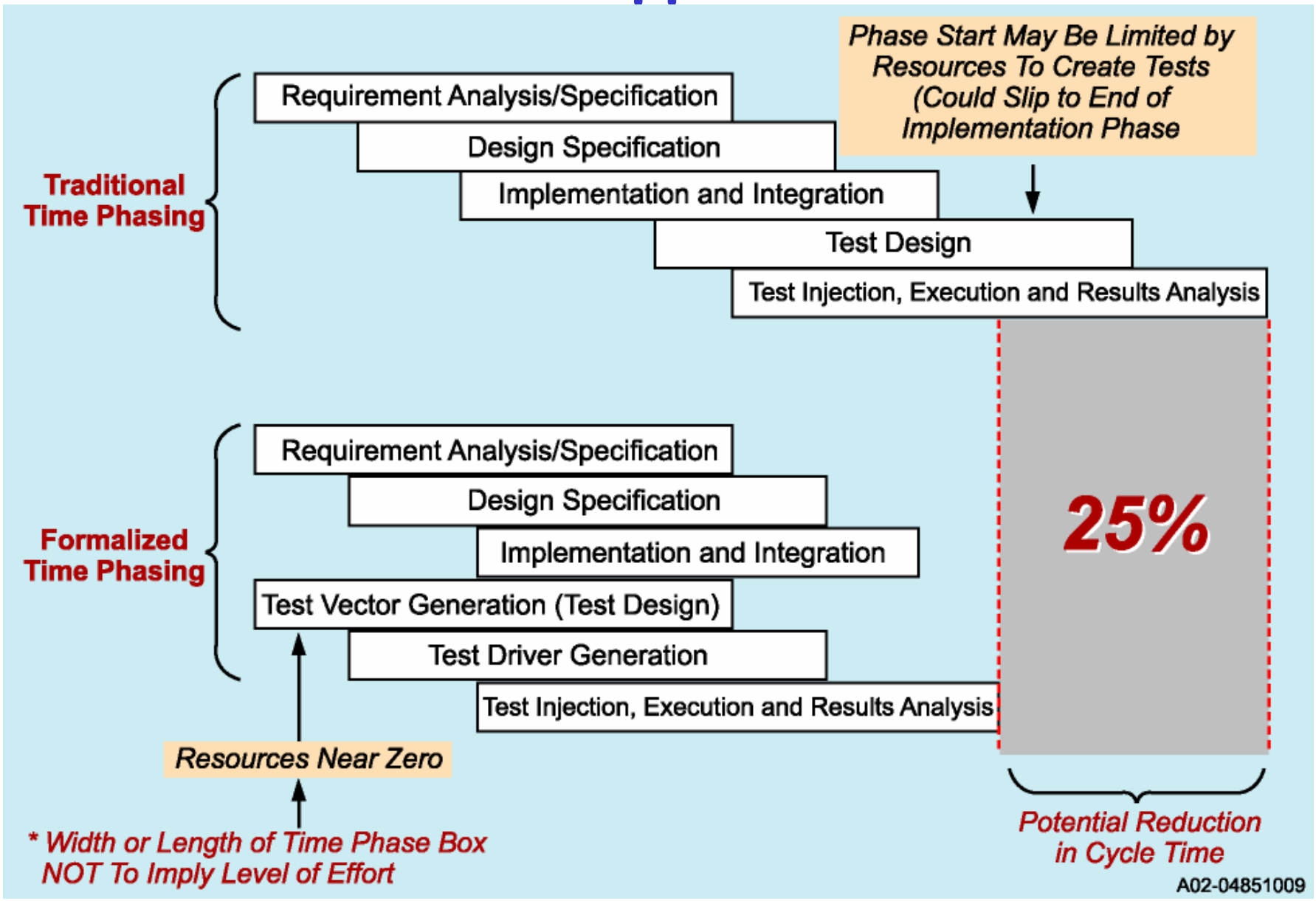




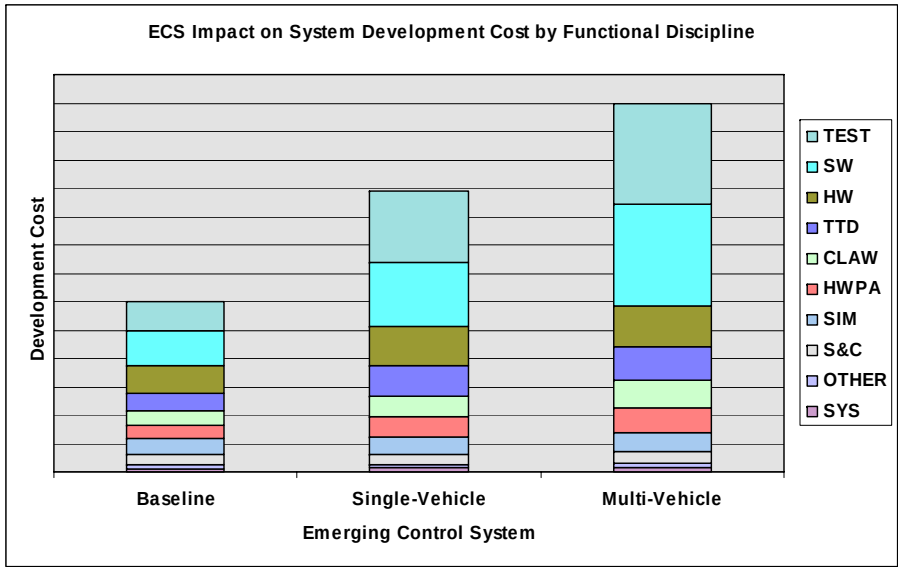
Future Military Program Testing Hours Are Forecast to Triple

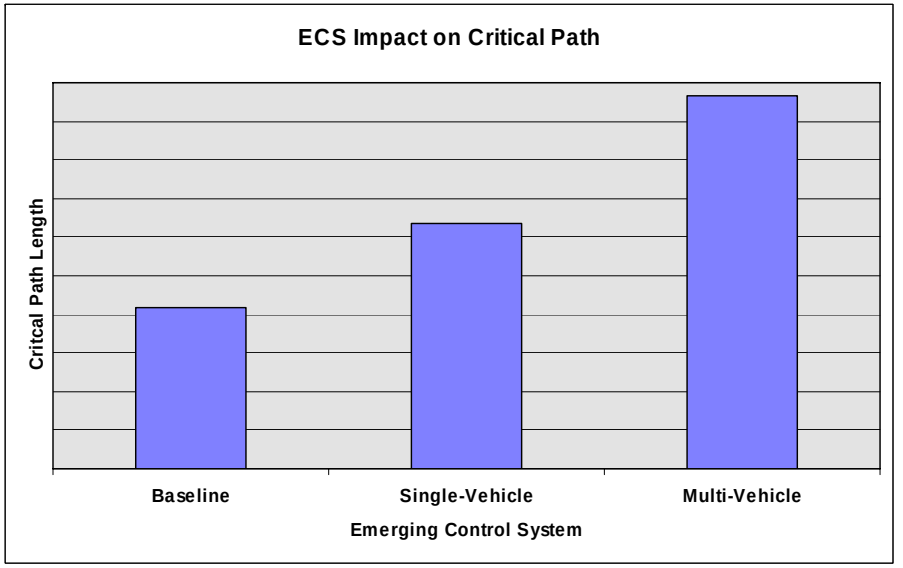


FCS – Possible applications of HSV



VVIACS - Impact Analysis Results





- **Single-Vehicle ECS Increases Development Costs ~ 50%, V&V Costs ~ 100%, and Critical Path Length ~ 50%**
- **Multiple-Vehicle ECS Increases Development Costs ~ 100%, V&V Costs ~ 150%, and Critical Path Length ~ 125%**
- **Software: Single-Vehicle 100% Increase and Multiple-Vehicle 200% Increase in V&V Costs**
- **Test: Single-Vehicle 150% Increase and Multiple-Vehicle 250% Increase in V&V Costs**

Significant Cost/Schedule Increase Projected Due to Complexity

Advanced Verification Strategy Evolutions

Near-Term (1-3 yrs) Evolution: System Model-based design now being implemented

- ***Auto-Code***
- ***Auto-Test***
- ***Rapid Prototyping***
- ***System Model-Based***
- ***Automated Verification Management***
- ***Simulation-Based design***

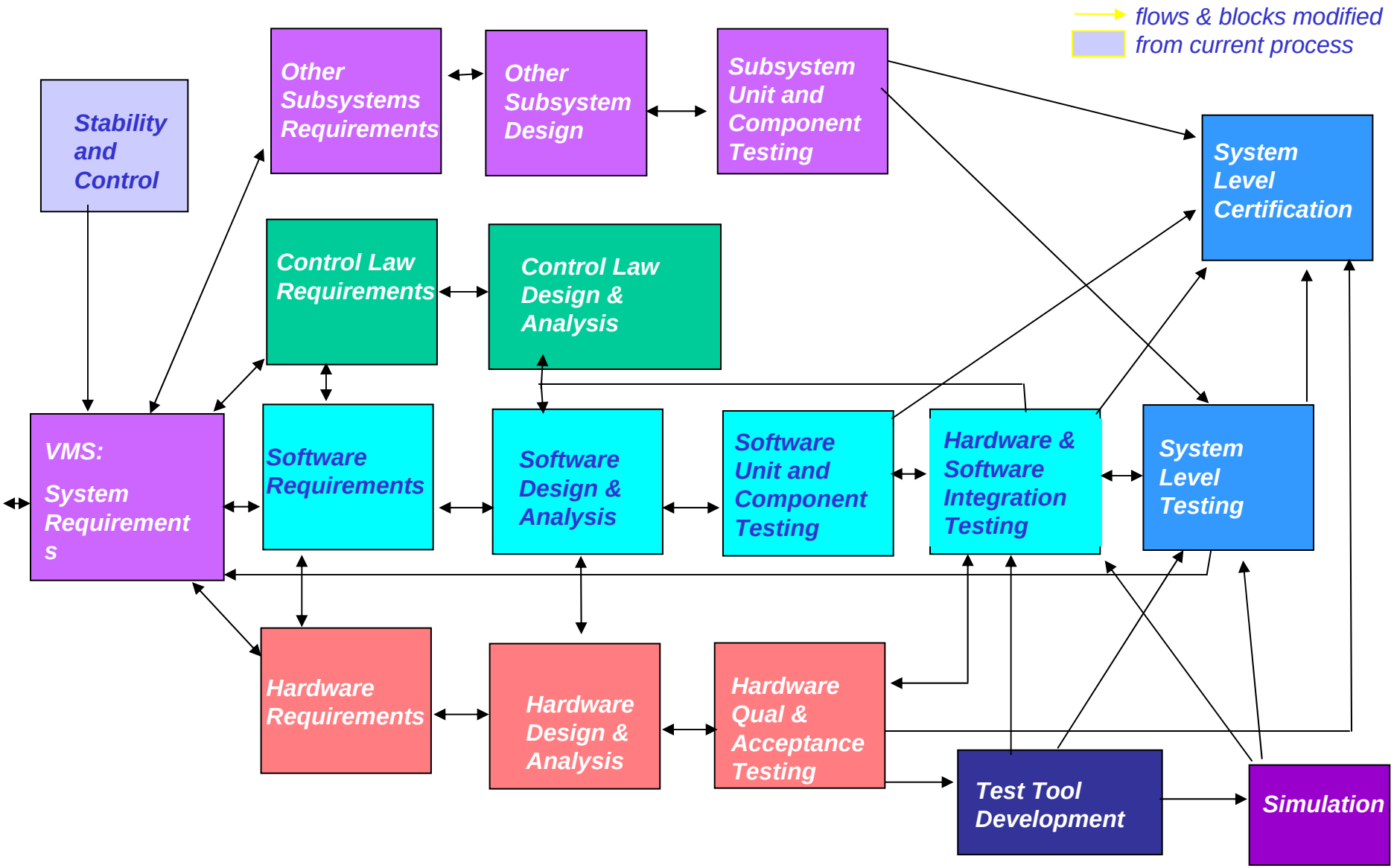
Mid-Term (4-6 yrs) Evolution: Formal Foundations in advanced development

- ***Formal Requirements Specs***
- ***Requirements and Traceability Analysis***
- ***Formal Methods***
- ***Computer-Aided System Engineering***

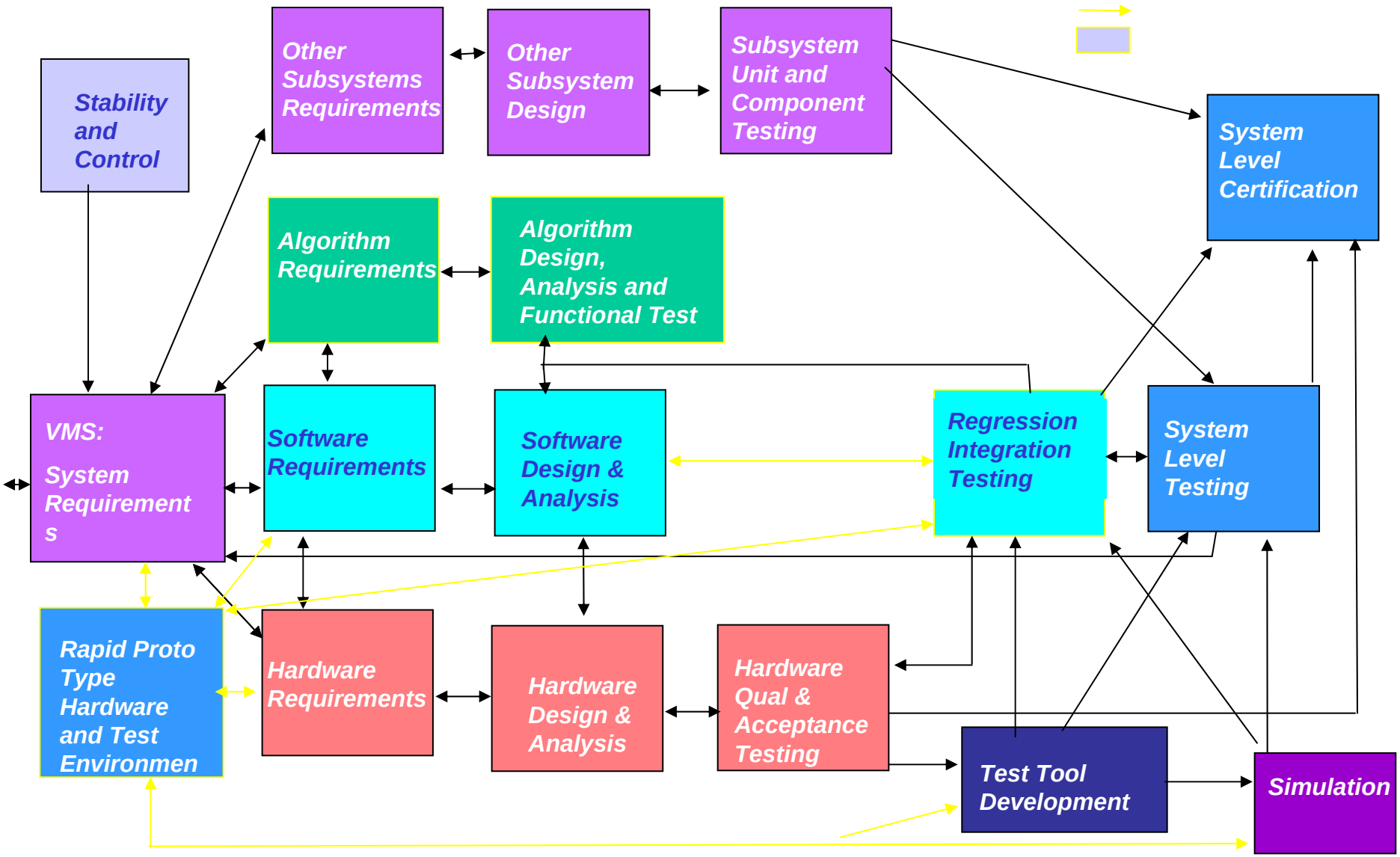
Far-Term (7-9 yrs) Evolution: V&V Awareness throughout – still in research

- ***V&V Run-time Design***
- ***Rigorous Analysis for Test Reduction***
- ***Requirements & Design Abstraction***
- ***Probabilistic/Statistical Test***
- ***Testing Metrics***

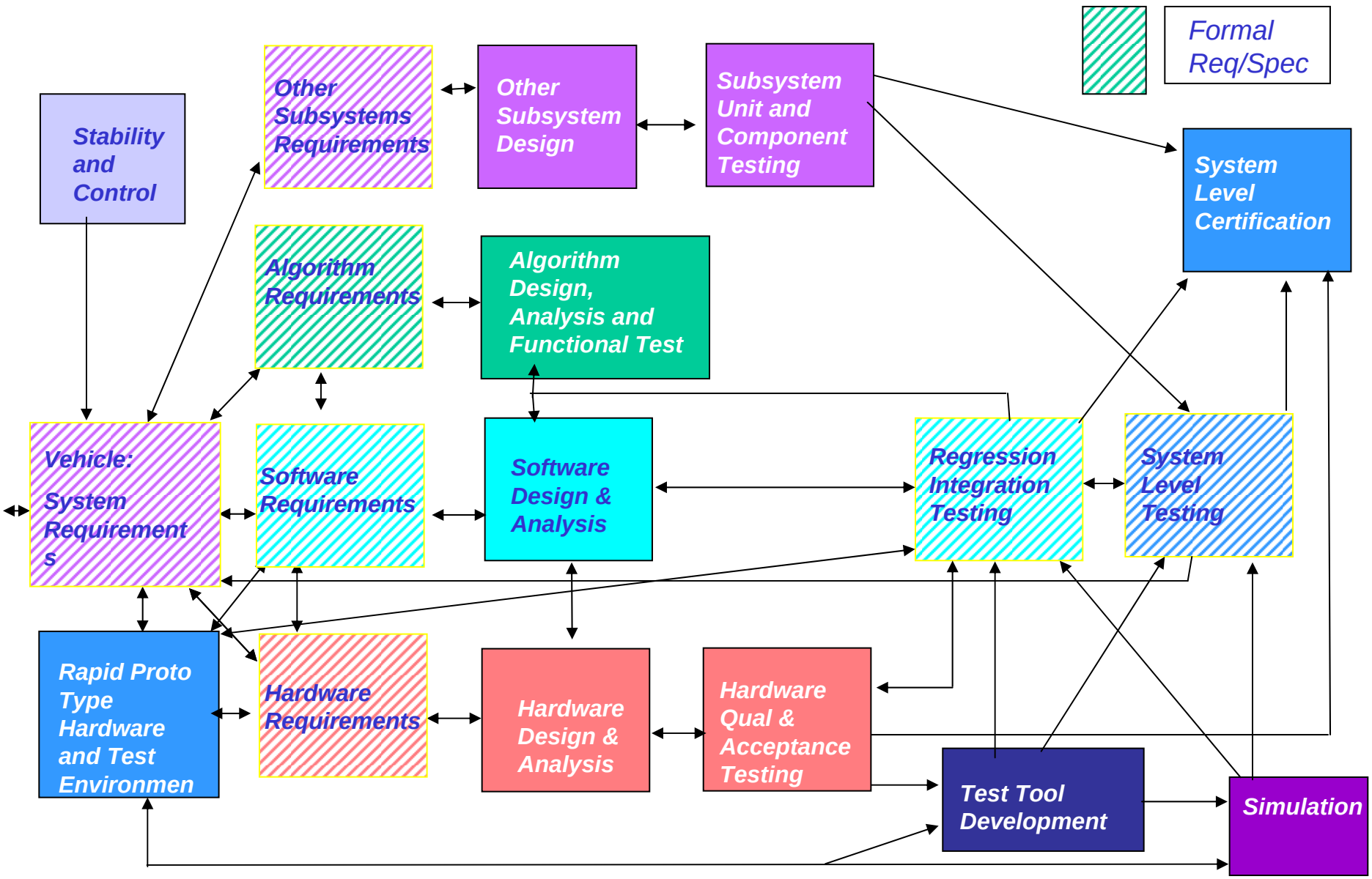
Current Process



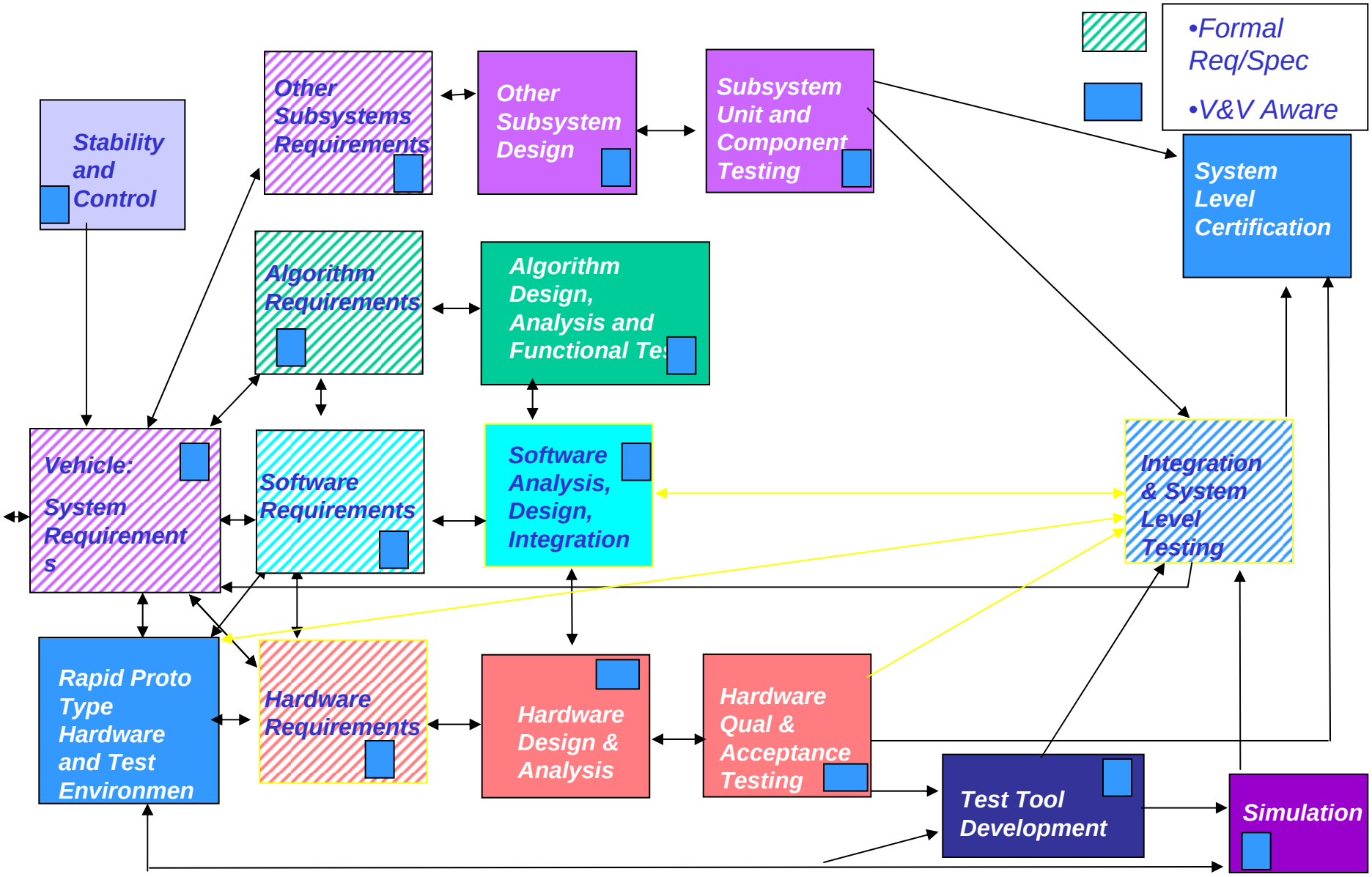
Near-Term (1-3 Yrs) Process



Mid-Term (4-6 Yrs) Process



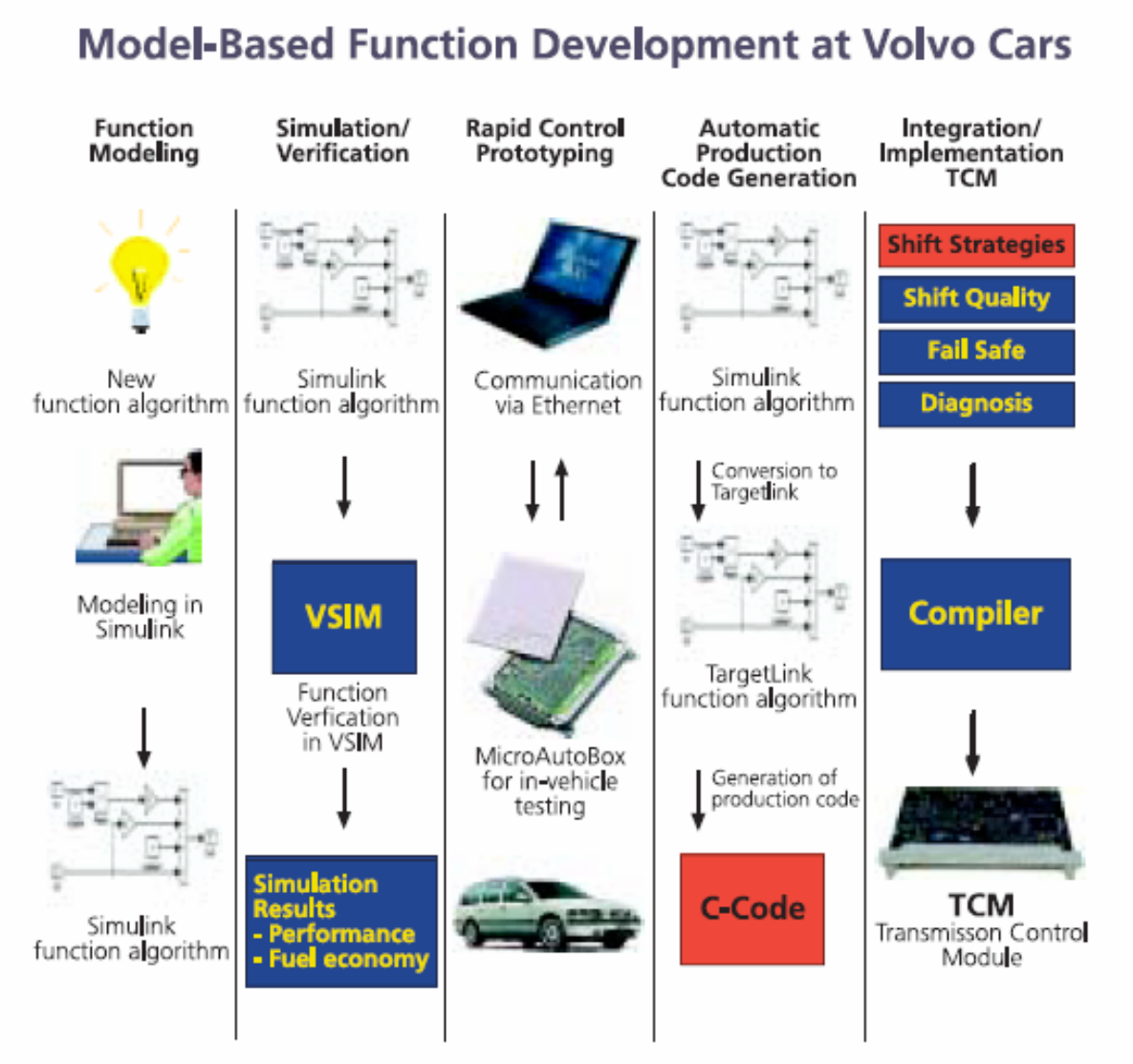
Far-Term (7-9 Yrs) Process



Flight Control Systems

- **Current status of HSV**
  - Formal methods are being used extensively in industry
  - Hybrid system tools are being explored by the industry
  - Considerable interest in changing the certification process
- **Barriers for HSV**
  - Sizes of the models
  - Integration with industry tools

Automotive Control Systems (ACS) – Design Flow

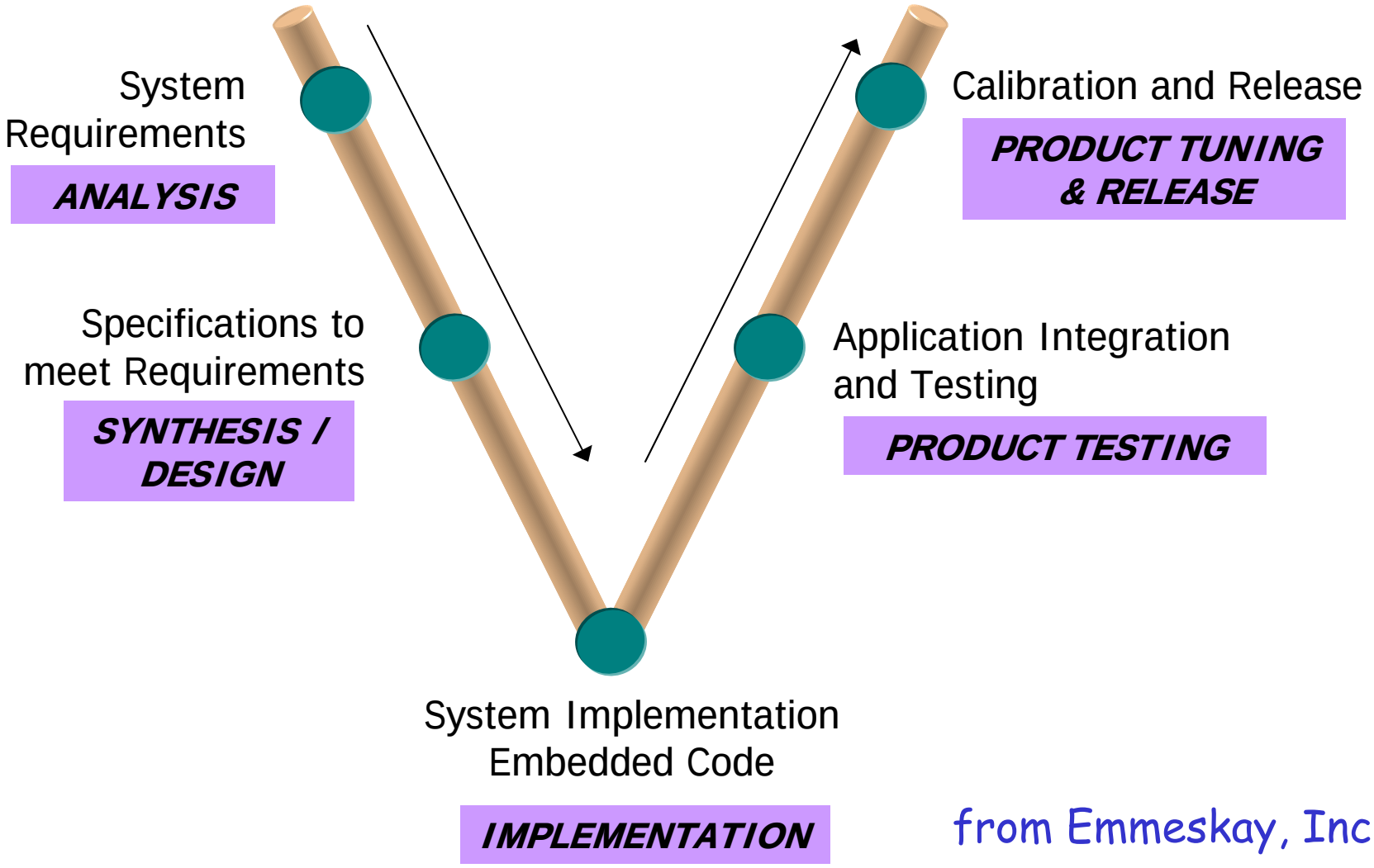


Goal:
Use models for

- requirements
- design
- verification
- test case generation
- modification
- documentation
- code generation
- run-time monitoring
- etc.

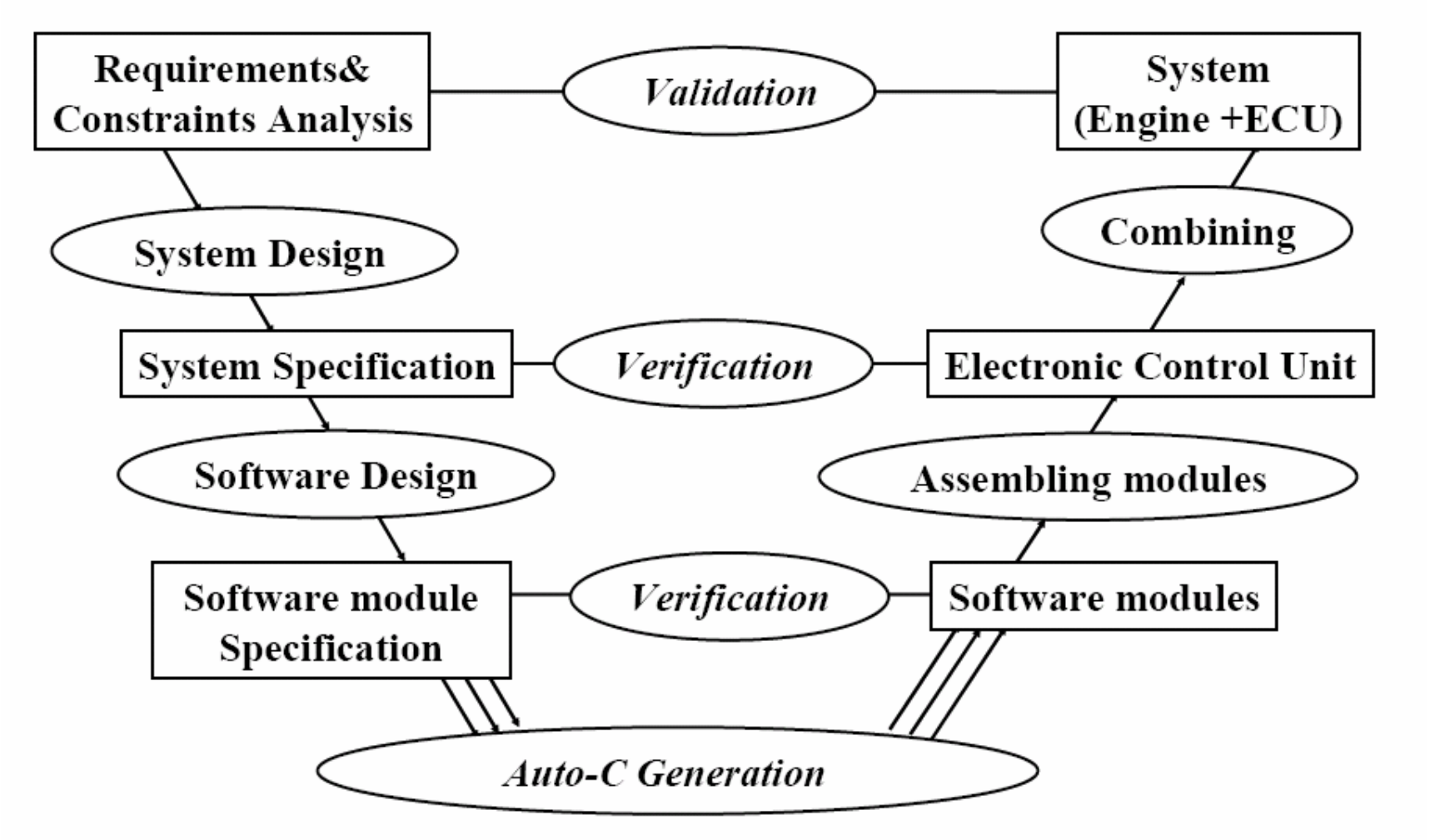
dSpace News, Jan. 02

Standard Embedded Software Design "V"



from Emmeskay, Inc.

ACS – Possible applications of HSV

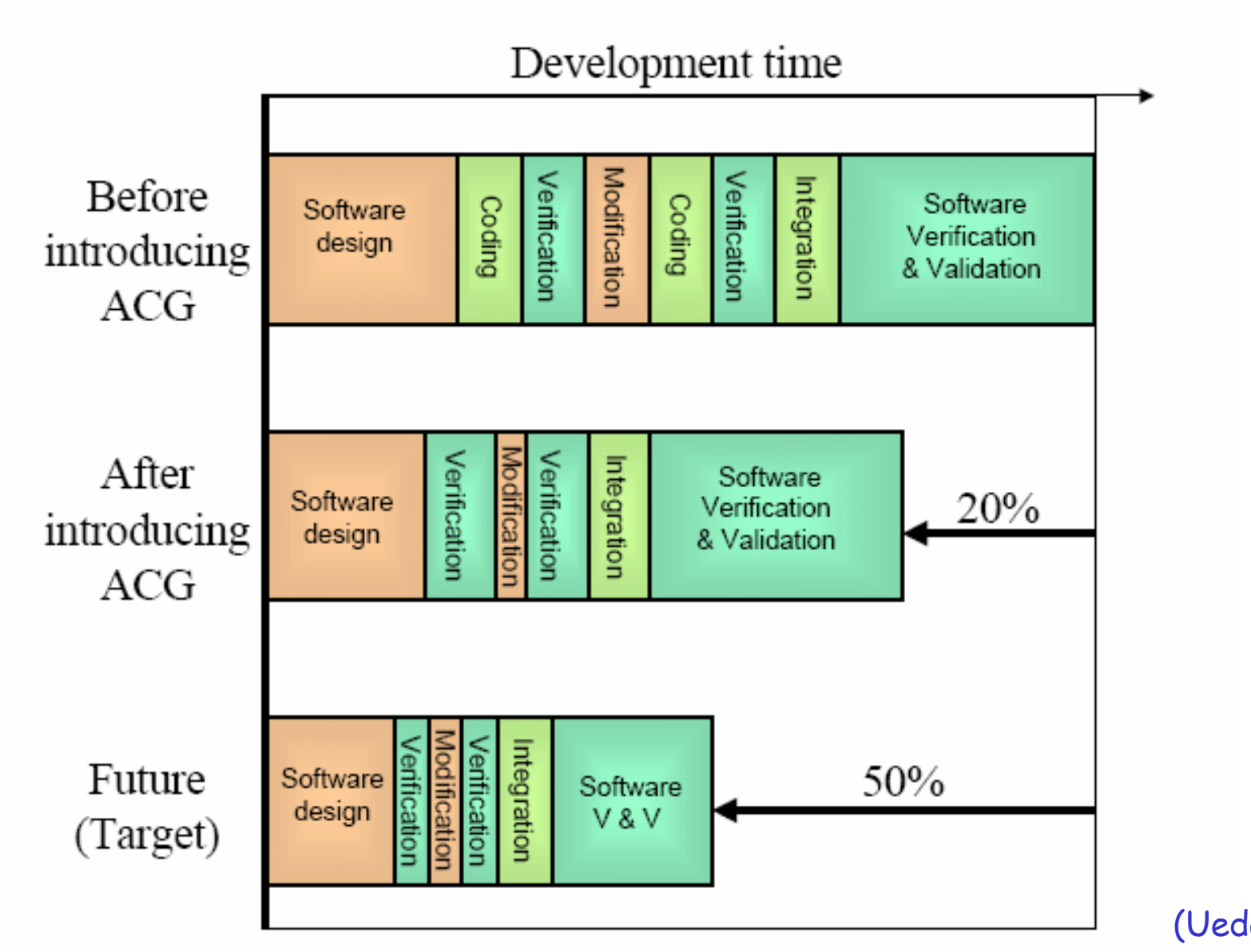


(Ueda & Ahata)

ACS – Current status of HSV

- Formal methods are being introduced for requirements capture and test-case generation
- Some exposure in industry
  - DARPA MoBIES project
  - Autmotive research labs have tried it
- Several demonstrations of concepts have been implemented
- Autocode is being used

Impact of Autocode (Toyota)



(Ueda & Ahata)

ACS – Barriers for HSV (Comments from industry)

- Requirements are not always understood or documented very well; humans are needed to interpret many verification results.
- The hybrid analysis tools we have seen to date are unable to:
  - Easily accept models already created and individually 'verified' from commercial tools like MATLAB or Matrix$_X$
  - Unable to deal with complexity of behaviors represented in our combined control and plant models
- It's not acceptable to have to create new models for verification – it takes too much time and is error prone. Need to work with the existing toolchain!

ACS – Barriers (Comments cont'd.)

- Software verification tools don't handle floating point variables
- Need semantic translation to move models between tools/languages.
- The industry uses tools that model physical systems, such as Matlab, MatrixX, Dymola. The verification tools need to work with these models.
- Need better education: need engineers who can grasp mechanical, computer, electrical and control ideas, as well as formal methods.
- Need robust verification -- not just verify the exact given model, but verify a set of neighborhood models because they are approximations and change in implementation

Technology Maturation Process

- Basic Research
- Concept Formation
- Development and Extension
- Internal Exploration
- External Exploration
- Popularization

Where are we at with hybrid system verification?

Critical Factors for Technology Acceptance (R&R)

- *Concept integrity* – no major outstanding questions
- *Clear recognition of need* – well-respected proponents
- *Tuneability* – fits a variety of technology user groups
- *Documented positive experience* – reports showing demonstrable attractive cost/benefit
- *Management commitment* –actively working to introduce the technology
- *Training* – target end users must be comfortable with the technology

Lessons Learned from Model Checking

- solutions must tailored to the domain
- stand-alone technologies are of limited value
- there must be champions of the technology in the industry
- researchers need nonproprietary test beds
- we need to be naively optimistic

When will we be able to verify real hybrid systems?

When will we be able to verify real hybrid systems?

Soon!

References

J. M. Buffington, V. Crum, B. H. Krogh, C. Plaisted, R. Prasanth, P. Bose, T. Johnson, Validation & verification of intelligent and adaptive control systems (VVIACS), *AIAA Guidance, Navigation and Control Conference*, Aug. 2004.

L. Fix and K. McMillan, Formal property verification, manuscript, 2006.

R. W. Floyd, *Assigning meanings to programs, Math. Aspects of CS, Proc. Symposia on Applied Math*, vol. 19, AMS, 1967, pp. 19-32.

G. Frehse, B.H. Krogh, R. Rutenbar, Verifying analog oscillator circuits using forward/backward abstraction refinement, DATE, 2006.

C.A.R. Hoare, An axiomatic basis for computer programming, Communications of the ACM, Jan 1983, pp. 53-56.

G.J. Holzmann. Design and validation of computer protocols, Prentice Hall, 1991.

R. P. Kurshan, Formal verification in a commercial setting, Design Automation Conference, 1997.

References

K. L. McMillan. Symbolic Model Checking: an approach to the state explosion problem, PhD Thesis. CMU CS-929131, 1992.

A. Ohata and K. Butts, Towards a concurrent engine system design methodology, 2005 American Control Convergence

A. Pnueli, The temporal logic of programs. In 18th Symposium on Foundations of Computer Science, pages 46-57, 1977.

A. Pnueli, The temporal semantics of concurrent programs, Theoretical Computer Science, 13:45-60, 1981.

V. Poladian, Software technology maturation study: model checking techniques and tools, Graduate Course Report, Penn State University, Fall 2001.

S. Redwine and W. Riddle, Software technology maturation, Proc. Eighth ICSE, May 1985, pp. 189-200.

T. Ueda and A. Ahata, Trends of future powertrain development and the evolution of powertrain control systems, Convergence Transportation Electronics Assocation, 2004 (best paper award, Convergence 2004).

References: Personal Communication

- Ken Butts, Toyota
- Paul Caspi, Verimag
- Bill Milam, Ford
- Oded Maler, Verimag
- Chris Meyers, University of Utah
- Pieter Mosterman, The MathWorks
- Rob Rutenbar, Carnegie Mellon University
- Shiva N. Sivashankar, Emmeskay, Inc.
- Mike Whalen, Rockwell Collins