

Addressing The Needs of Real-Time Embedded Software A Case for Software Systems Engineering

Rob Pettit

Flight Software and Embedded Systems Office

The Aerospace Corporation

Adjunct Professor of Computer Science

George Mason University and Virginia Tech

Motivation

- Software plays a critical role in our everyday lives
 - *Software, rather than hardware is now the dominant force in the control of embedded devices*
 - Hand-held phones and MP3 players to mission/life critical control systems
 - *For space systems, software is the root cause for at least half of observed anomalies*
- Trends in computer science and software engineering have actually moved away from supporting the reliable development of real-time embedded systems
 - *Processes adopted based more on popularity than to the application domain*
 - *Loss of rigor in the application of development methods*
 - *Dependence on increasing hardware resources*



Addressing the Deficiencies

- Software Systems Engineering
 - *Software engineers grounded in systems engineering concepts*
 - Application of computer science to large-scale software systems
 - Understanding of hardware/platform level interactions with software
 - *Should also education systems engineers on software engineering*
- Solid foundations are specifically needed for:
 - *Architecture*
 - *Performance*
 - *Predictability*
 - *Efficiency*
 - *Model-based methods*



Conclusions

- As reliance on software controlled systems continues to increase, so will the size and complexity of the software
- Software systems engineering curriculum is critical to the success of large-scale, real-time, and embedded software systems
 - *Must provide rigorous approaches to software development that are scalable for industrial applications*
 - *Must continue to advance software engineering research*
 - But also provide clear criteria for adopting new approaches

