



University of Southern California

Center for Systems and Software Engineering

Educating T-Shaped Computer Science Students

Hardware Peopleware Economics Applications Disciplines

S
o
f
t
w
a
r
e

Barry Boehm, University of Southern California

GSAW 2018, Feb. 28, 2018



Personal USC Agenda

- **Simon Ramo: Our best TRW engineers are T-shaped**
 - Strong in at least one technical discipline
 - With working knowledge of other success-critical disciplines
- **But most of our software new-hires were I-shaped CS grads**
- **Early retirement decision: try to create T-shaped SW-engineers**
 - USC MS-CS with specialization in software engineering
 - Key courses include software architecture, user interaction, software test & analysis, software management & economics
 - And 2-semester real-client project course (2000 students to date)
 - Foundation-stone rather than capstone
 - Clients generally unfamiliar with software technology
 - Fall semester focused on software-intensive systems engineering
 - Operations concept, Winbook requirements negotiation, prototyping, architecture, life cycle plans, compatibility and feasibility evidence



Factors contributing to I-shaped software engineer problems

- An increasing number of new computer science (CS) degree programs fill up CS students' schedules, leaving little room for non-CS courses providing skills outside of CS. CS “breadth courses” are more CS courses.
- Hardware-first system engineering practices often discourage software engineers to participate in system engineering activities
- Narrow-focused Software-CMM (Capability Maturity Model) provided further discouragement. Here is KPA 1 (Rqts Engr), Activity 1:
 - Analysis and allocation of the system requirements is not the responsibility of the software engineering group but is a prerequisite for their work



Example problems created by I-shaped software engineers

- **The Golden Rule:** Do unto others as you would have others do unto you, i.e., build programmer-friendly user interfaces for doctors,
 - **Platinum Rule:** Do unto others as they would be done unto.
- **Computer scientists prize abstraction**
 - **User name:** U1, U2 vs Jim, Tina
 - Inventing personas effectively helps students represent classes of stakeholders
- **Making programmer-convenient, but user-inconvenient decisions**
 - 10-day data buckets vs weekly, monthly reporting



T-shaped MSCS-SwEngr degree program

- **Foundation-stone real-client project course**
- **Software Management and Economics**
- **User Interface Design and Development**
- **Hardware-Software Embedded Systems**
- **Systems and Software Architecting**
 - **Using Rechtin Systems Architecting approach**
- **Later courses in agile methods, software verification and validation, systems and software requirements**



Foundation-Stone Course Practices (1/2)

Joint with Sue Mobasser while at USC

- Visit clients' workplace and jointly develop a desired concept of operation
- Jointly negotiate prioritized stakeholder win-win requirements
- Jointly develop evaluation criteria for choices of non-developmental items
- Jointly determine and prioritize project risks, develop risk mitigation plans
- Develop clients' business case linking investments to quantitative and qualitative benefits



Foundation-Stone Course Practices (2/2)

- Identify complementary client activities
- Participate in 4 major milestone reviews with clients and instructors
- Develop initial increment and hold a client Core Capability Drivethrough
- Jointly negotiate prioritized end-game revisions
- Transition software and support materials



Resulting Student Benefits

Hiring organizations come back for more

- **CS students need more than CS skills to survive in an inter-disciplinary world**
- **With T-shaped curriculum, students can**
 - **Build up their job interview portfolio**
 - **Acquire non-outsourcable skills**
 - **Have a better understanding of hiring manager needs**
 - **Come up a rapid assimilation curve, and**
 - **Learn how to learn**
- **Rapid changes in technology make systems-oriented software engineers critical to the success of most future system developments**