University of Southern California
Center for Software Engineering

Skating to Where the Puck Is Going:
Future-Proofing Opportunities and Challenges

Barry Boehm, USC-CSSE
http://csse.usc.edu

Keynote Address, STC 2013
April 9, 2013

Motivation

- "What helped me most in becoming a good hockey player was learning to skate to where the puck was going, rather than where it was or where it had been."
  - Wayne Gretzky, NHL Hall of Fame
- "Reflection in action," or asking, "How could we have done our last project better?" is actually skating to where the puck has been.
  - It is very valuable, but needs to be balanced with anticipating the future
  - For example, don't stop at CMMI Level 4

Outline

- ⇨ The Future of Information Technology
  - 8 surprise-free trends; 2 wild-card trends
  - Changes since 2006 paper
  - Individual and combined software engineering opportunities and challenges
- Conclusions: Future-Proofing
  - Research, acquisition, staffing/education

The Future of Systems and Software: 2006

Best paper of 2006, Systems Engineering journal

- Eight surprise-free trends
  1. Increasing integration of SysE and SwE
  2. User/Value focus
  3. Software Criticality and Dependability
  4. Rapid, Accelerating Change
  5. Distribution, Mobility, Interoperability, Globalization
  6. Complex Systems of Systems
  7. COTS, Open Source, Reuse, Legacy Integration
  8. Computational Plenty
- Two wild-card trends
  9. Autonomy Software
  10. Combinations of Biology and Computing

2013 Trends Largely Missed in 2006

- Nanotechnology megasensor-intensive smart systems
- Search and mining of ultralarge data aggregations
- Software implications of multicore chips
- Rapid growth of cloud computing, service-orientation
- Rapid growth of social networking technologies

The Future of Systems and Software: 2013

- ⇨ Eight surprise-free trends
  1. Rapid, Accelerating Change
  2. Software Criticality and Dependability
  3. Complexity; Global/Mobile Systems of Systems
  4. COTS, Open Source, Services, Legacy Integration
  5. Smart Systems; Mining huge volumes of data
  6. User Evolution and End Value Focus
  7. Computational Plenty and Multicore Chips
  8. Increasing integration of SysE and SwE
- Two wild-card trends
  9. Autonomy Software
  10. Combinations of Biology and Computing

University of Southern California
Center for Software Engineering

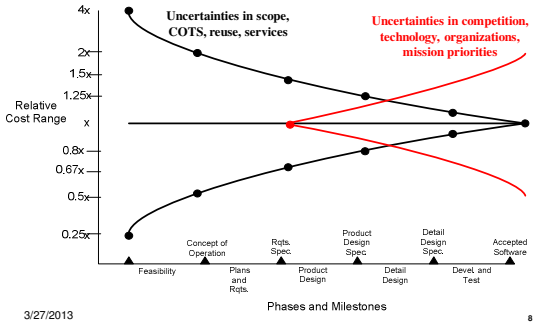
1. Rapid Change Trends

- Global connectivity and competition accelerate change
  - More ripple effects of technology, marketplace changes
- Increased need for agility, continuous learning
  - Need to balance agility and plan-driven dependability
  - Decline of THWADI (That's how we've always done it)
  - Avoid technical agility, administrative THWADI
- Hybrid agile/plan-driven processes needed for larger systems
- Need for incremental processes, methods, tools, skills
- Need for pro-active technology, marketplace monitoring
- Education: Need to learn how to learn

University of Southern California
Center for Software Engineering

Adaptation Challenges: A Dual Cone of Uncertainty

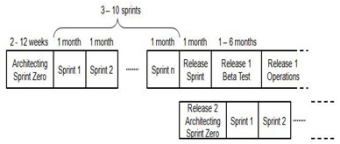
– Need early and later systems engineering, evolutionary development



University of Southern California
Center for Software Engineering

Architected Agile Approach

- Uses Scrum of Scrums approach
  - Up to 10 Scrum teams of 10 people each
  - Has worked for distributed international teams
  - Going to three levels generally infeasible
- General approach shown below
  - Often tailored to special circumstances



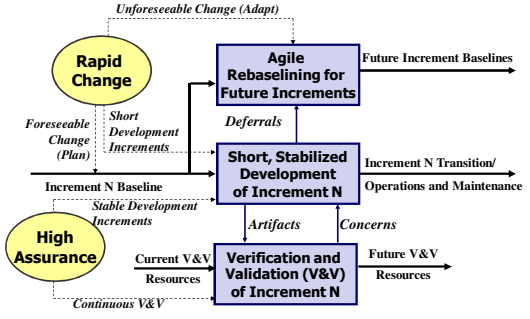
University of Southern California
Center for Software Engineering

2. Criticality and Dependability Trends

- Software increasingly success-critical to product and service systems
  - Provides competitive differentiation, adaptability to change
- Dependability is generally not vendors' top-priority
  - "The IT industry spends the bulk of its resources… on rapidly bringing products to market." – US PITAC Report
- By 2025, there will be a "9/11" – magnitude software failure
  - Major loss of life or collapse of world financial system
- This will raise dependability to vendors' top priority
  - Market demand; stronger warranties and accountability
  - Value-based system dependability processes and tools
    - Reflect all stakeholders' value dependencies

University of Southern California
Center for Software Engineering

Evolutionary Concurrent Engineering: Incremental Commitment Spiral Model



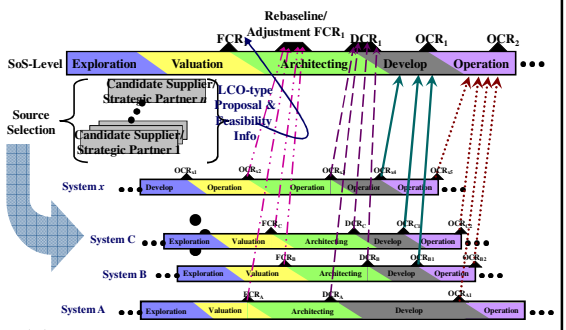
University of Southern California
Center for Software Engineering

3. Complexity and Global Software-Intensive Systems of Systems (SISOS)

- Lack of integration among stovepiped systems causes
  - Unacceptable delays in service
  - Uncoordinated and conflicting plans
  - Ineffective or dangerous decisions
  - Inability to cope with fast-moving events
- Increasing SISOS benefits
  - See first; understand first; act first
  - Network-centric operations coordination
  - Transformation of business/mission potential
  - Interoperability via Integrated Enterprise Architectures

University of Southern California
Center for Software Engineering

Future DoD Challenges: Systems of Systems



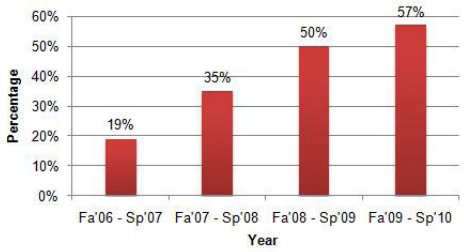
University of Southern California
Center for Software Engineering

The Future of Systems and Software: 2011

- **Eight surprise-free trends**
  1. Rapid, Accelerating Change
  2. Software Criticality and Dependability
  3. Complexity; Global/Mobile Systems of Systems
  4. ⇨ COTS, Open Source, Services, Legacy Integration
  5. Mining huge volumes of data
  6. User Evolution and End Value focus
  7. Computational Plenty and Multicore Chips
  8. Increasing integration of SysE and SwE
- **Two wild-card trends**
  9. Autonomy Software
  10. Combinations of Biology and Computing

University of Southern California
Center for Software Engineering

4. Purchased Services (Cloud Computing) Growth in USC e-Services Projects

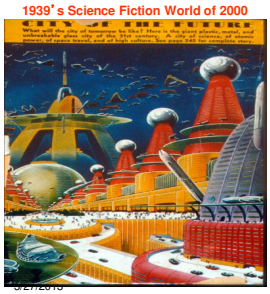


University of Southern California
Center for Software Engineering

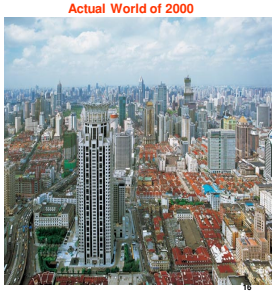
Persistence of Legacy Systems

- Before establishing new-system increments
  - Determine how to undo legacy system

1939's Science Fiction World of 2000



Actual World of 2000



University of Southern California
Center for Software Engineering

5. Megasensor- Empowered Smart Systems

Smart power grids, buildings, companies, cities

- Ubiquitously-instrumented artifacts and processes
- Complementary growth in data storage and analysis
- EU Digital Agenda "Internet of Things"
- Commitments: Singapore, Abu Dhabi, S. Korea, Portugal
  - Industry: IBM, HP, Cisco, Siemens, GE
- Generally Greenfield; incrementally for Brownfield

University of Southern California
Center for Software Engineering

Mining huge volumes of data

- Google example: billions (B) of search hits
  - All in about 0.3-0.5 seconds (2/17/2013)
  - Video: 13.8B; Time: 11.4B; News: 10.6B; Music: 8.2B; Life: 6.2B; Play: 5.4B
  - What to show first?
  - How to narrow search to what you want?
- Recommender systems
  - Based on preference data or past activity
  - Amazon.com; Pandora; Netflix
- Service-provider data warehousing
  - Better services, but service provider has your data
- General concerns with privacy, controls

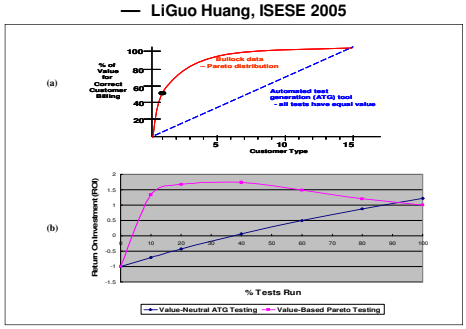
6. User/Value Focus Trends

- Computerworld panel: More focus on user/ownership costs and benefits; less focus on features and license costs
  - Technology should adapt to people, not vice versa
  - Tension between usability and feature creep
- User-orientation has many challenges
  - Emergent needs and priorities: IKIWISI, Maslow
  - Diversity of people and cultures: no OSFA solutions
  - Group vs. individual performance
  - Golden Rule: Do unto others as you would have others do unto you
    - Engineer focus on engineer-usability
    - Platinum Rule: Do unto others as they would be done unto

IKIWISI: I'll know it when I see it OSFA: one size fits all

Value-Based Testing: Empirical Data and ROI

— LiGuo Huang, ISESE 2005



The Future of Systems and Software: 2010

- Eight surprise-free trends
  1. Rapid, Accelerating Change
  2. Software Criticality and Dependability
  3. Complexity; Global/Mobile Systems of Systems
  4. COTS, Open Source, Services, Legacy Integration
  5. Mining huge volumes of data
  6. User Patterns and End Value Focus
  7. ⇨ Computational Plenty and Multicore Chips
  8. Increasing integration of SysE and SwE
- Two wild-card trends
  9. Autonomy Software
  10. Combinations of Biology and Computing

7. Computational Plenty and Multicore Chips

- Moore's Law stymied by heat dissipation problems
  - 2x circuit speed, density every 18 months
- Keep growth by developing multi-CPU chips
  - Lower circuit speed, but lower power consumption
  - Growth in #CPUs keeps up processing power growth
  - But only if programs can be parallelized
  - Otherwise, legacy software will run more slowly
  - Amdahl's Law: Speed limited by speed of slowest part on critical computation path
- But can also use CPUs for other purposes
  - Assertion checking, intrusion detection, trend analysis, option analysis, performance monitoring, fault tolerance, one-pass Monte Carlo

8. Increasing SysE/SwE Integration

- Can't do good SwE by neglecting SysE
  - Weak SysE the root cause of most SW project failures
- Can't do good SysE by neglecting critical success factors
  - Software an increasing system critical success factor
    - Provides most of competitive differentiation
    - Provides most of adaptability to change
    - Enables later binding of commitments

9, 10. Wild Cards: Autonomy and Bio-Computing

- Great potential for good
  - Robot labor; human shortfall compensation
    - ❖ 5 Senses, healing, life span, self-actualization
  - Adaptive control of the environment
  - Redesigning the world for higher quality of life
    - ❖ Physically, biologically, informationally
- Great potential for harm
  - Loss of human primacy: computers propose, humans decide
  - Overempowerment of humans
    - ❖ Accidents, terrorism, 1984 revisited
  - New failure modes: adaptive control instability, self-modifying software, commonsense reasoning, bio-computer mismatches
  - V&V difficulties: cooperating autonomous agents, biocomputing
- Forms and timing of new capabilities still unclear

University of Southern California Center for Software Engineering

Outline

- The Future of Information Technology
  - 8 surprise-free trends; 2 wild-card trends
  - Changes since 2006 paper
  - Individual and combined software engineering opportunities and challenges

⇨ Conclusions: Future-Proofing
  - Research, acquisition, staffing/education

University of Southern California Center for Software Engineering

Future-Proofing Research Needs

- Incremental vs. start-over tools and methods (e.g., for formal methods)
- Ambiguity-tolerant methods, processes, and tools
- Powerful, flexible, composable models and simulations
- Rapid change-impact analysis
- Scalable methods, processes, and tools
- Concurrent engineering support
  - Collaboration technology; quality attribute tradeoff analysis tools
  - Value-based architecting and design methods, processes, and tools
- Continuous V&V; early, continuous testing; reliable autonomy
- Use of multi-core computing processors for rapid multiple-option analysis
- Integration of hardware, software , and human factors solution approaches and architectures
- Next-generation process maturity models
- Evolutionary acquisition contract and incentive structures
- Rapid technology maturation and adoption

University of Southern California Center for Software Engineering

Future-Proofing Processes

- Evolutionary vs. single-step processes
- No clear boundary between "development," "operations and maintenance"
  - Don't release your key SE people at PDR
- Short prioritized increments; Schedule as independent variable (SAIV)
  - Architect to add or drop borderline-priority features to meet schedule
  - Exceptions for non-subsettable deliverables
- Earlier testing, verification, and validation (V&V)
  - Architecture, infrastructure V&V to avoid unscalable easiest-first increments
  - Earlier/continuous test, V&V
- Rapid, concurrent, pro-active SE and acquisition management
  - Of product and process; requirements and solutions; development and evolution; hardware/software/human factors
  - Proactive vs. reactive with respect to technology, NDI, external interfaces
- Evidence/risk as decision criterion, first class deliverable
  - Process for evidence preparation
  - Risk/opportunity-driven: avoid overkill, underkill (balance risk, opportunity)
  - Competitive prototyping as a way of buying information to reduce risk

University of Southern California Center for Software Engineering

Rapid, Adaptive Acquisition Management

- Need to rework traditional guidance documents
  - No longer prespecified, total-package, sequential IMP, IMS , WBS, EVMS
  - Concurrently engineered with evidence-based decision milestones
- Include evolution requirements along with capability, interface, level of service, project requirements
- No "one-size-fits-all" contracting
  - SE budgets a function of system size, criticality, requirements volatility
  - Award fees rewarding collaborative, effective change adaptation
- Continuous monitoring: INCOSE leading indicators; SERC effectiveness measures, or equivalent
  - Evidence preparation; schedule; exit criteria
  - Success-critical stakeholders participation; validation by experts
- Acquisition corps empowerment
  - Next generation of tools, education, and training; career paths

University of Southern California Center for Software Engineering

Evidence-Based Reviews

- *Evidence* provided by developer and *validated by independent experts* that:

If the system is built to the specified architecture, it will

- Satisfy the specified operational concept and requirements
  - Capability, interfaces, level of service, and evolution
- Be buildable within the budgets and schedules in the plan
- Generate a viable return on investment
- Generate satisfactory outcomes for all of the success-critical stakeholders
  - Shortfalls in evidence are uncertainties and risks
- Should be resolved or covered by risk management plans
  - Assessed in increasing detail at major decision milestones
- Serves as basis for stakeholders' commitment to proceed
- Serves to synchronize and stabilize concurrently engineered elements

Can be used to strengthen current schedule- or event-based reviews

University of Southern California Center for Software Engineering

SysE and SwE Education Implications

- Current SysE and SwE students will be practicing into the 2050s. Their education should consider the following:
  - Anticipating future trends and preparing students to deal with them;
  - Capitalizing on information technology to enable the delivery of just-in-time and web-based education;
  - Monitoring current principles and practices and separating timeless principles from outdated practices;
  - Participating in leading-edge software engineering research and practice and incorporating the results into the curriculum;
  - Packaging smaller-scale educational experiences in ways that apply to large-scale projects;
  - Helping students learn how to learn, through state-of-the-art analyses, future-oriented educational games and exercises, and participation in research; and
  - Offering lifelong learning opportunities for systems engineers who must update their skills to keep pace with the evolution of best practices