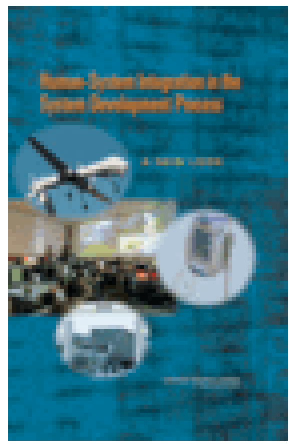
Human-System Integration in the System Development Process: A New Look



Frank E. Ritter

with some slides from Barry Boehm

21 July 09

Goals of the Tutorial

- Introduce the report
- Explain the report
- Learn how to leverage the results of the report
- Teach and be taught about system design
- Discuss the application of user models that it represents
- See new issues and problems and opportunities
- Provide you with tools to argue for better system design and to reduce risk

Audience for the Tutorial

- HSI professionals
- Designers
- Modelers
- HCI

Glossary

- System
  collection of different elements that produce results not obtainable by elements alone
- System of Systems
  Originally defined for own purposes, are combined and coordinated to produce a new system
- BDUF - Big design up front (re: BUFF)
- PDR - Product Requirements Document
- Risks - situations or events that cause projects to fail to meet goals
- ICM - incremental commitment model
- LSI - Lead system integrator
- LCO - life cycle objectives
- LCA - Life cycle architecture
- IOC - initial operating capability

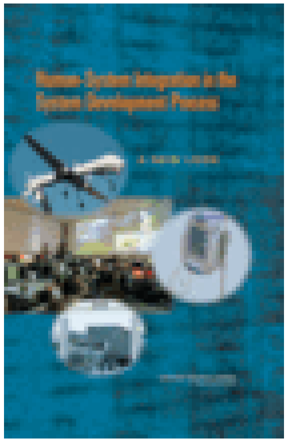
Problems with (Future) Systems of Systems Development

- Lack of commitment by funders, managers to avoid HSI risks
- Lack of communication between system engineers and human-system experts
- Difficulties providing data about humans into the design process
- Thus, the study/literature survey at beginning of book
  (also see Booher & Minniger, 2003)

Parts

- Part 0: Preamble [1-6]
- Part 1: Overview [7-23]
- Part 2: Stages [24-37]
- Part 3: Managing risks [38-46] [Break?]
- Part 4: HSI methods [47-57]
- Part 5: Summary [58-

Pew and Mavor (2007) Charged to:



- Working with a panel, to
- Comprehensively review issues
- Evaluate state of the art in human-system integration (and engineering)
- Develop a vision
- Recommend a research plan

COMMITTEE ON HUMAN-SYSTEM DESIGN SUPPORT FOR CHANGING TECHNOLOGY

RICHARD W. PEW (*Chair*), BBN Technologies, Cambridge, MA
NIGEL BEVAN, University of York, London
BARRY W. BOEHM, Computer Science Department, University of Southern California
NANCY J. COOKE, Department of Psychology, Arizona State University
SHELLEY EVENSON, School of Design, Carnegie Mellon University
DAVID GRAEBER, Boeing Phantom Works, Seattle, WA
EDMOND W. ISRAELSKI, Abbott Laboratories, Abbott Park, IL
BRIAN M. KLEINER, Grado Department of Industrial and Systems Engineering, Virginia Polytechnic Institute
MICHAEL MULLER, IBM Research, Cambridge, MA
FRANK E. RITTER, College of Information Sciences and Technology, Pennsylvania State University
EMILIE ROTH, Roth Cognitive Engineering, Brookline, MA
THOMAS F. SANQUIST, Battelle Seattle Research Center, Pacific Northwest National Laboratory, Seattle, WA

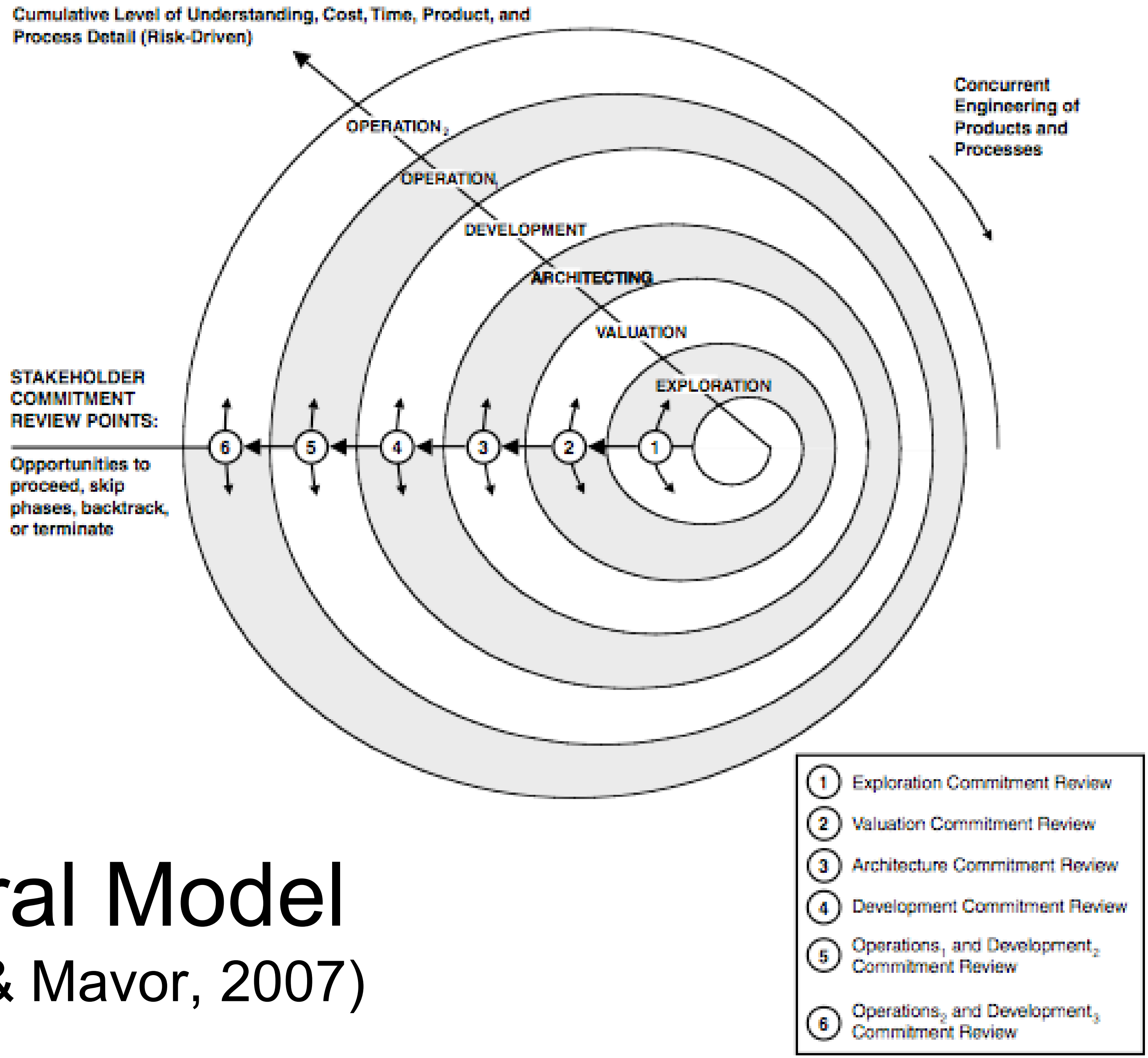
ANNE S. MAVOR, *Study Director*
KRISTEN A. BUTLER, *Research Assistant*
MATTHEW D. McDONOUGH, *Senior Program Assistant*

Some Principles of System Development

- Satisficing
- Incremental growth
- Iterative development
- Concurrent system definition and development
- Management of project risk

Life cycle phases

- Exploration
- Valuation
- Architecting
- Development
- Operation



Spiral Model
(Pew & Mavor, 2007)

Essentials of the Spiral Model

(Boehm & Hansen, 2001)

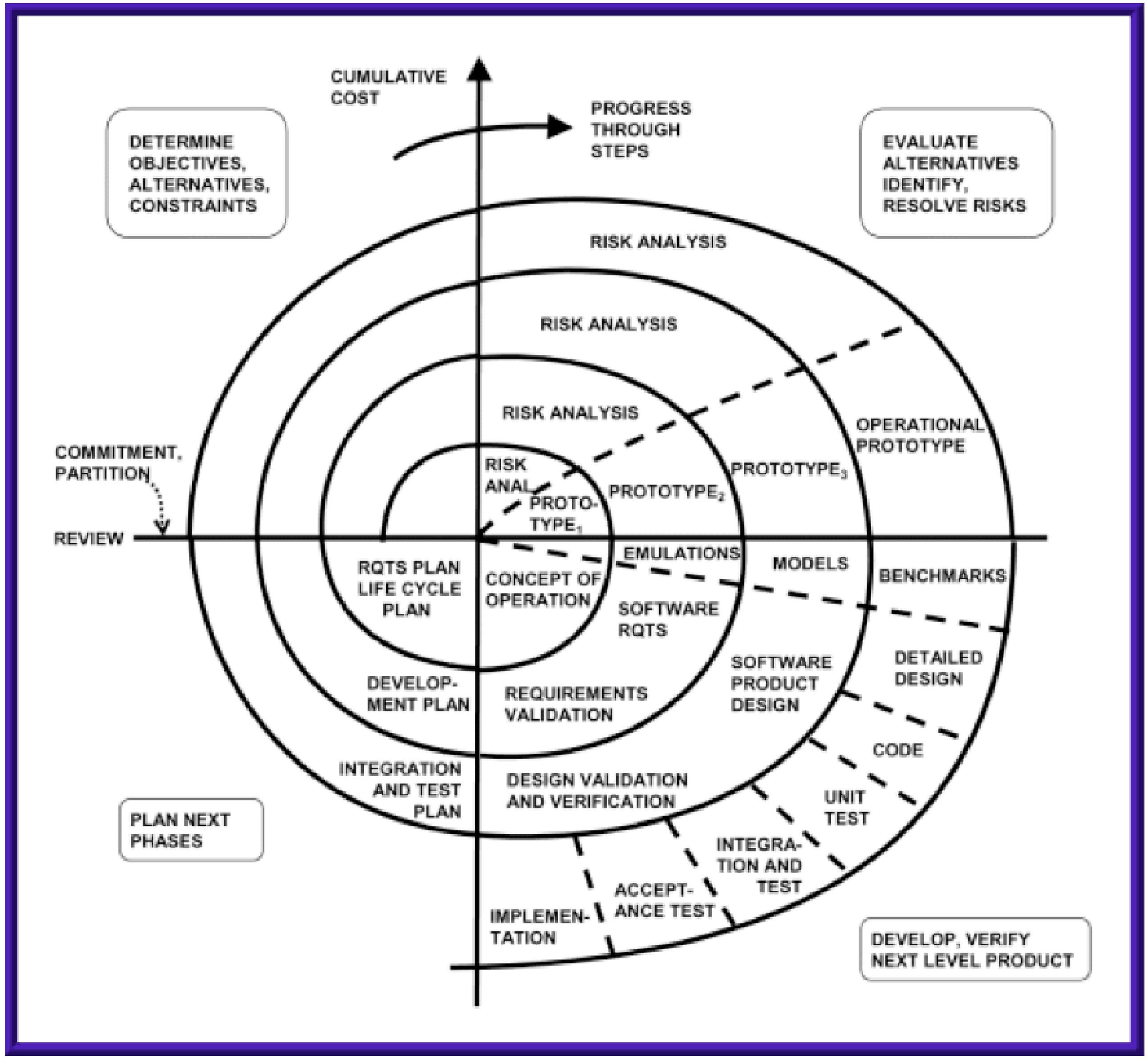
- Concurrent development of key artifacts
- Each cycle does Objectives, Constraints, Alternatives, Risks, Review, and Commitment to Proceed
- Level of effort driven by risk
- Degree of detail driven by risk
- Use anchor point milestones
- Emphasis on system and life cycle activities and artifacts

Incremental Commitment in Gambling

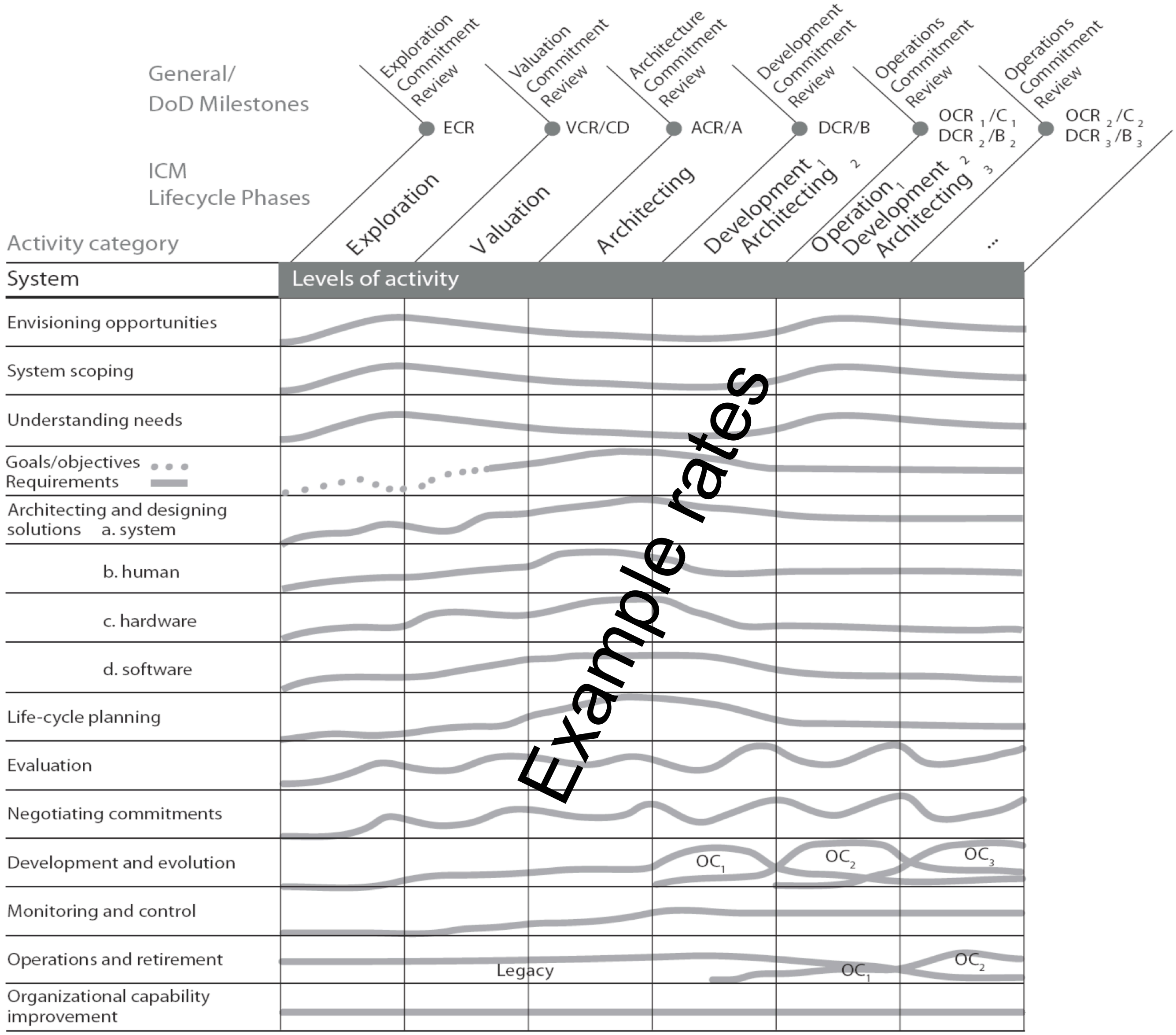
- Total Commitment: Roulette
  - Put your chips on a number
    - E.g., a value of a key performance parameter
  - Wait and see if you win or lose
- Incremental Commitment: Poker, Blackjack
  - Put some chips in
  - See your cards, some of others' cards
  - Decide whether, how much to commit to proceed

Spiral Model

(Boehm & Hansen, 2001)



ICM HSI Levels of Activity for Complex Systems



Example implication: Less testing!

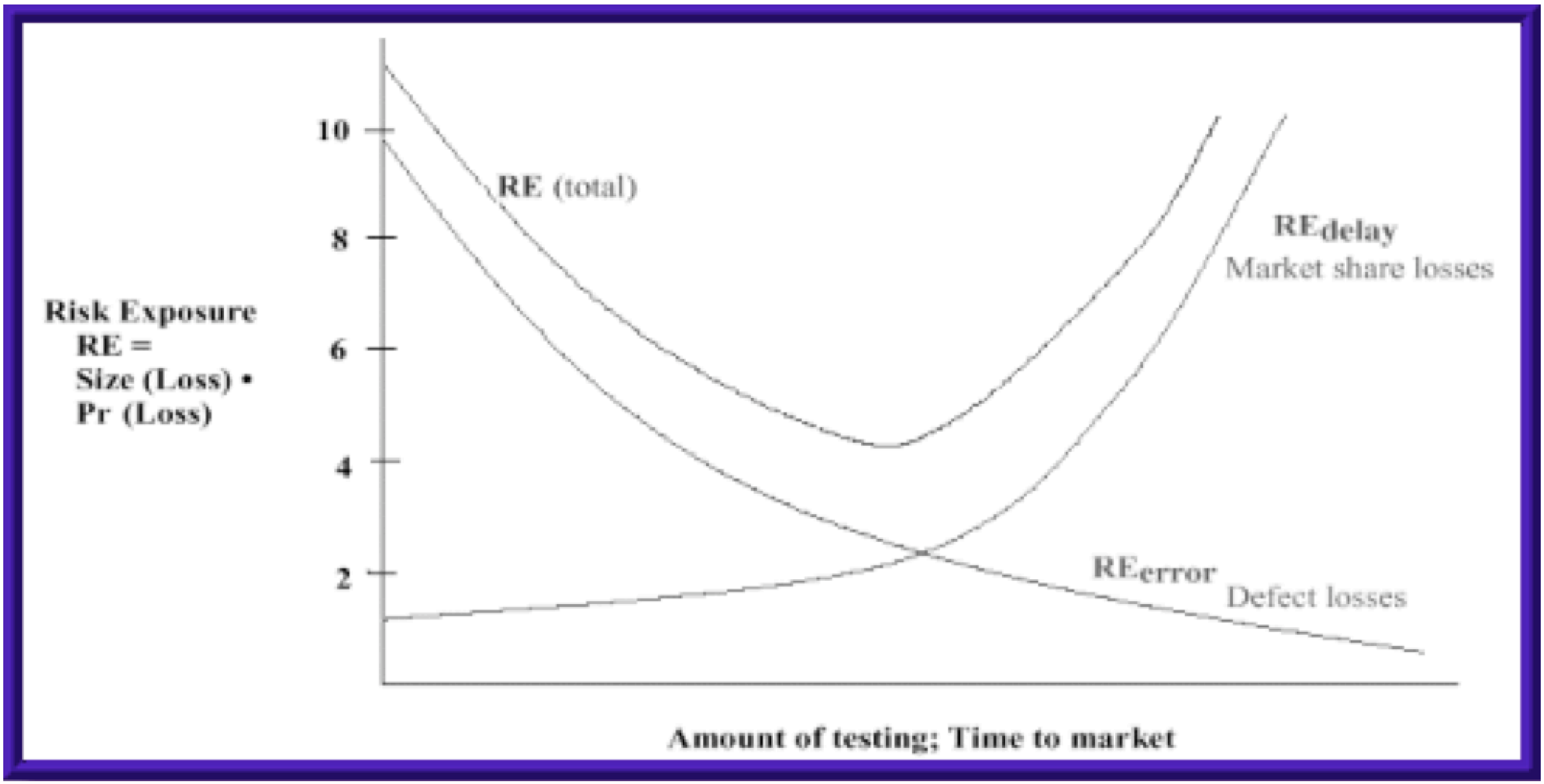


TABLE 2-1 Principles-Based Comparison of Alternative Process Models

	Principles				
Process Models	Stakeholder Satisficing	Incremental Growth	Concurrency	Iteration	Risk Management
Sequential waterfall, V	Assumed via initial requirements; no specifics	Sequential	No	No	Once at the beginning
Iterative, risk-driven waterfall, V	Assumed via initial requirements; no specifics	Risk-driven; missing specifics	Risky parts	Yes	Yes
Risk-driven evolutionary development	Revisited for each iteration	Risk-driven; missing specifics	Risky parts	Yes	Yes
Concurrent engineering	Implicit; no specifics	Yes; missing specifics	Yes	Yes	Implicit; no specifics
Agile	Fix shortfalls in next phase	Iterations	Yes	Yes	Some
Spiral process 2001	Driven by stakeholder commitment milestones	Risk-driven; missing specifics	Yes	Risk-driven	Yes
Incremental commitment	Stakeholder-driven; stronger human factors support	Risk-driven; more specifics	Yes	Yes	Yes

Process Model Principles

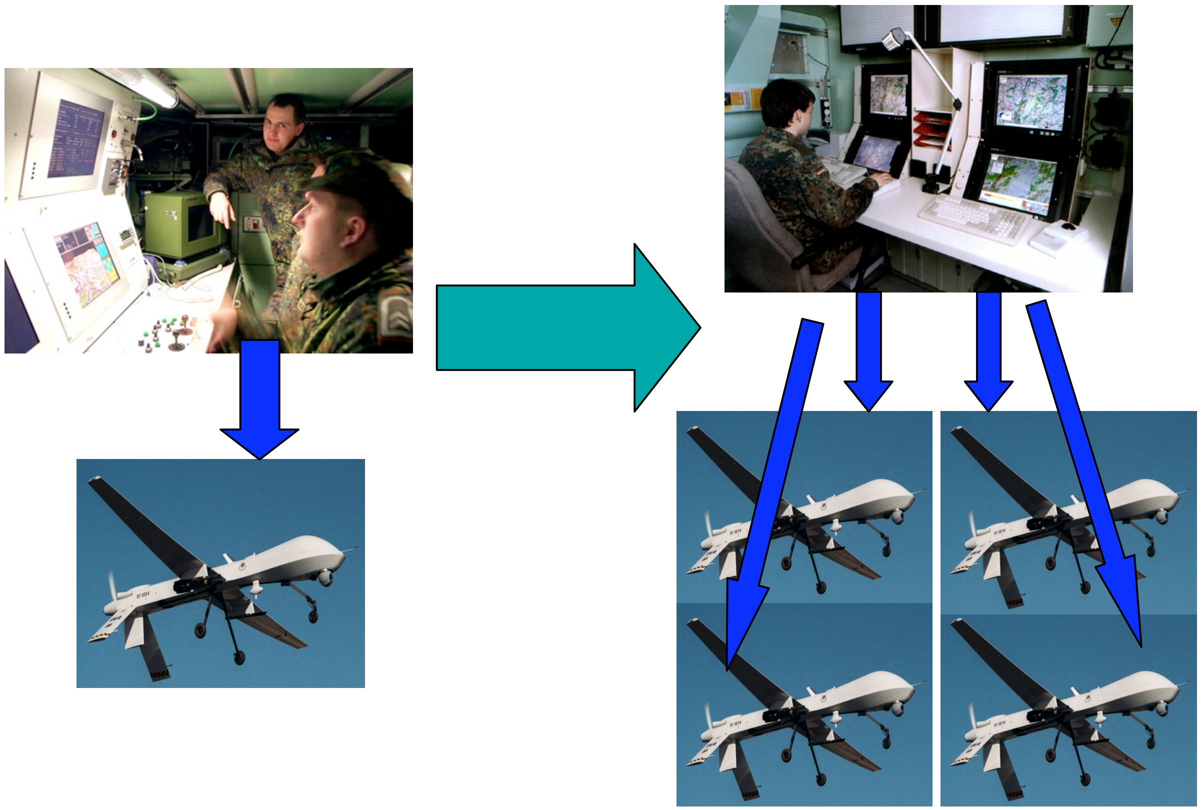
1. Commitment and accountability
2. Success-critical stakeholder satisficing
3. Incremental growth of system definition and stakeholder commitment

4, 5. Concurrent, iterative system definition and development cycles

    Cycles can be viewed as sequential concurrently-performed phases or spiral growth of system definition

6. Risk-based activity levels and anchor point commitment milestones

Small example: Scalable remotely controlled operations 1 of 2



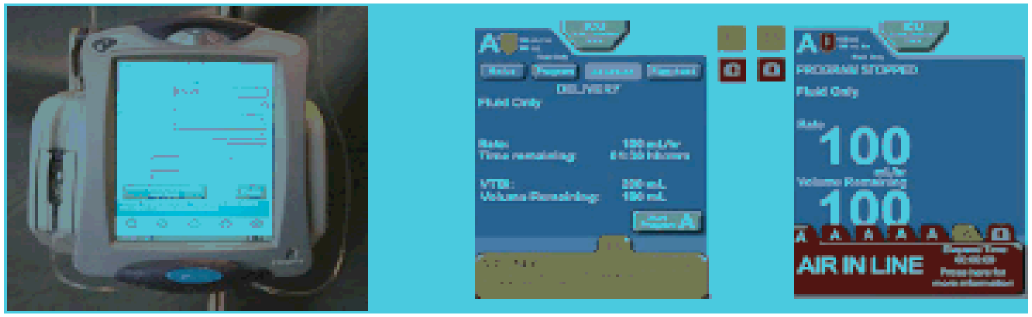
Total vs. Incremental Commitment – 4:1 RemPilotVeh 2 of 2

- Total Commitment
  - Agent technology demo and PR: Can do 4:1 for $1B
  - Winning bidder: $800M; PDR in 120 days; 4:1 capability in 40 months
  - PDR: many outstanding risks, undefined interfaces
  - $800M, 40 months: "halfway" through integration and test
  - 1:1 IOC after $3B, 80 months
- Incremental Commitment [number of competing teams]
  - $25M, 6 mo. to VCR [4]: may beat 1:2 with agent technology, but not 4:1
  - $75M, 8 mo. to ACR [3]: agent technology may do 1:1; some risks
  - $225M, 10 mo. to DCR [2]: validated architecture, high-risk elements
  - $675M, 18 mo. to IOC [1]: viable 1:1 capability
  - 1:1 IOC after $1B, 42 months

Example ICM HCI Application: Symbiq Medical Infusion Pump

Winner of 2006 HFES Best New Design Award

Described in NRC HSI Report, Chapter 5



Symbiq IV Pump ICM Process - I

- Exploration Phase
  - Stakeholder needs interviews, field observations
  - Initial user interface prototypes
  - Competitive analysis, system scoping
  - Commitment to proceed
- Valuation Phase
  - Feature analysis and prioritization
  - Display vendor option prototyping and analysis
  - Top-level life cycle plan, business case analysis
  - Safety and business risk assessment
  - Commitment to proceed while addressing risks

Symbiq IV Pump ICM Process - II

- Architecting Phase
  - Modularity of pumping channels
  - Safety feature and alarms prototyping and iteration
  - Programmable therapy types, touchscreen analysis
  - Failure modes and effects analyses (FMEAs)
  - Prototype usage in teaching hospital
  - Commitment to proceed into development
- Development Phase
  - Extensive usability criteria and testing
  - Iterated FMEAs and safety analyses
  - Patient-simulator testing; adaptation to concerns
  - Commitment to production and business plans

Implications of RD-ICM

- Comparable to waterfall
  (see http://www.waterfall2006.com/)
- Risks related to humans are often ignored by system engineers
- Risks related to hardware are ignored by HF professionals
- *People naturally work on risks*
  so theory is not just normative but descriptive
- See recommendations in book
- *Can/could/should bring in experts to advise*
- Others?

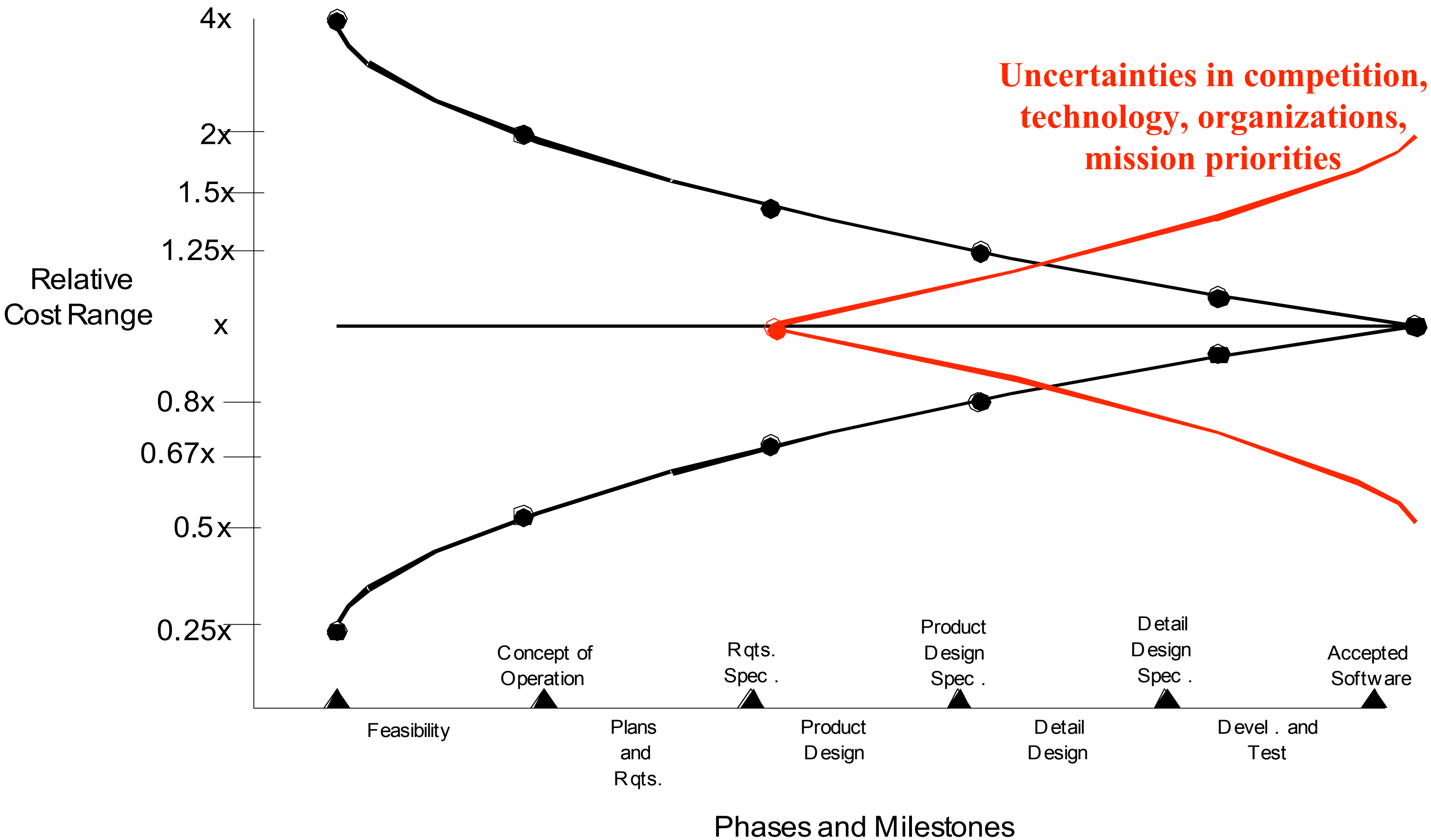
Part 2: Looking at Stages of the Process

TABLE 2-2 Primary Focus of HSI Activity Classes and Methods

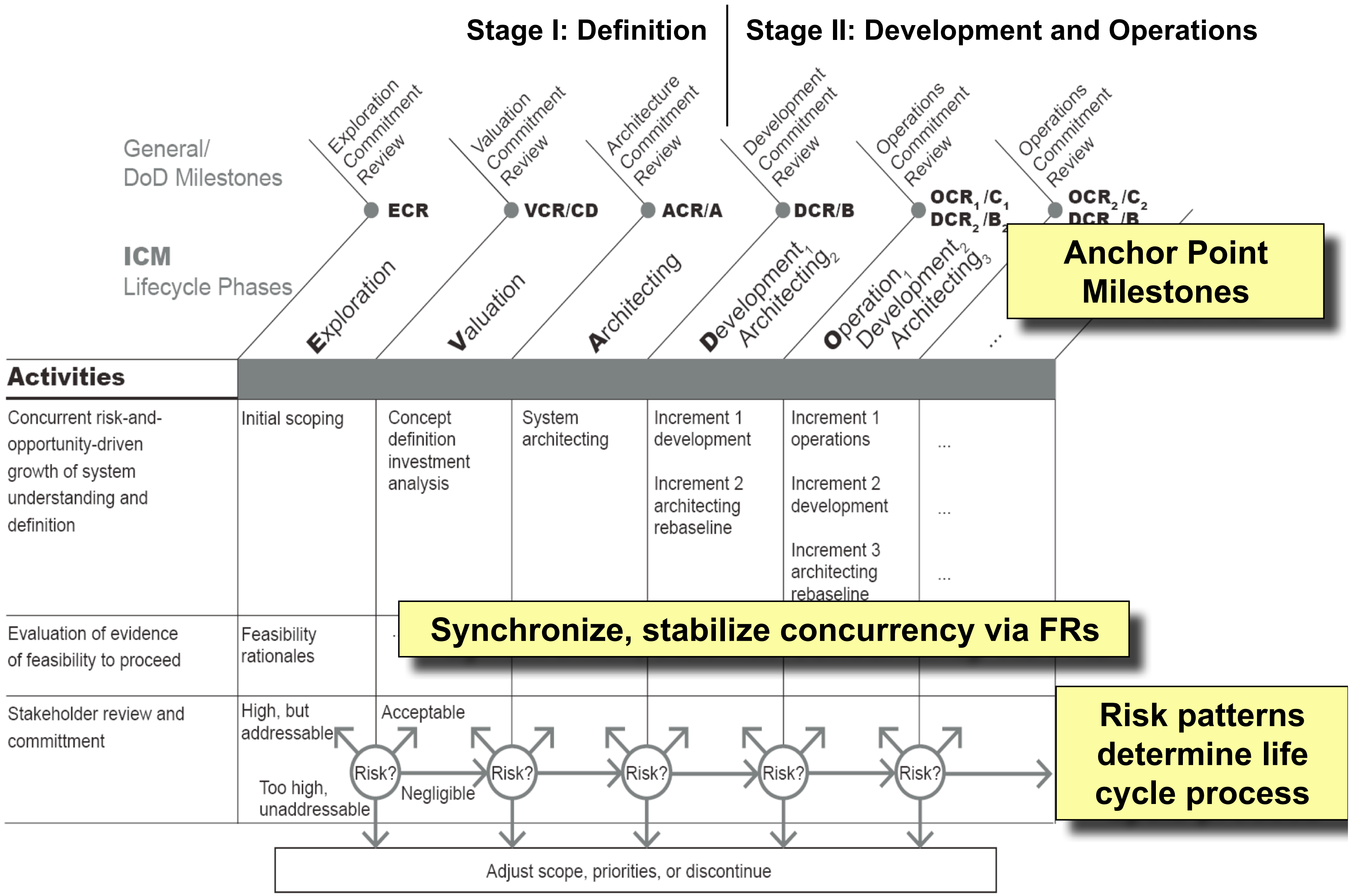
Activity Class	Examples of HSI Methods Described in This Volume	Systems Engineering
1. Envisioning opportunities	–Field observations and ethnography –Participatory analysis	–Modeling –Change monitoring (technology, competition, marketplace, environment)
2. System scoping	–Organizational and environmental context analysis –Field observations and ethnography –Participatory analysis	–Investment analysis –System boundary definition –Resource allocation –External environment characterization –Success-critical stakeholder identification
3. Understanding needs	–Organizational and environmental context analysis –Field observations and ethnography –Task analysis –Cognitive task analysis –Participatory analysis –Contextual inquiry –Event data analysis –Prototyping –Models and simulations –Usability evaluation methods	–Success-critical stakeholder requirements –Competitive analysis –Market research –Future needs analysis

There is Another Cone of Uncertainty:

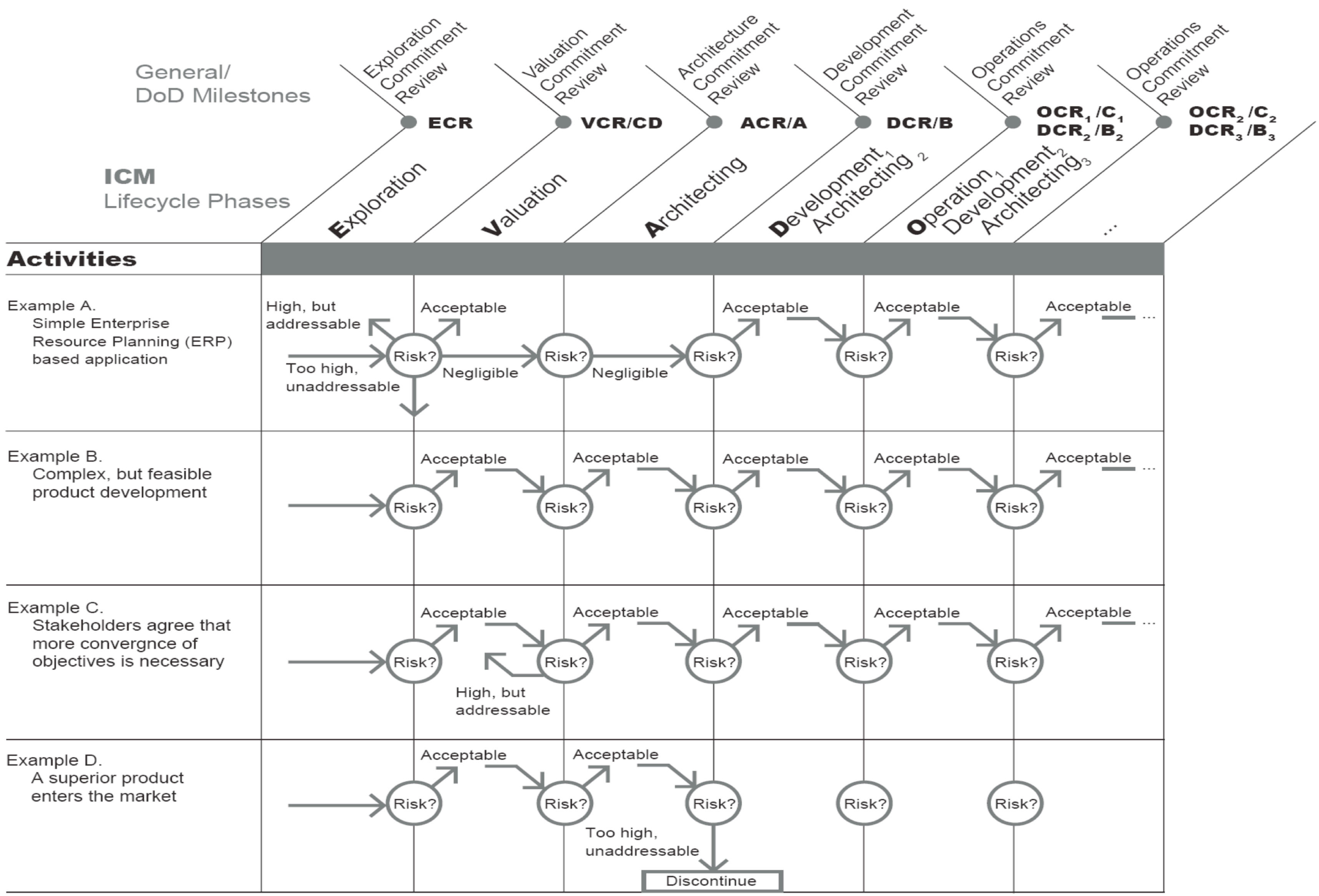
Shorter increments are better



The Incremental Commitment Life Cycle Process: Overview



Different Risk Patterns Yield Different Processes



ICM Assessment

- ICM principles and process are not revolutionary
- They repackage current good principles and practices to make it easier to:
  - Determine what kind of process fits your project
  - Keep your process on track and adaptive to change
- And harder to:
  - Misinterpret in dangerous ways [if you address all risks]
  - Gloss over key practices
  - Neglect key stakeholders and disciplines
  - Avoid accountability for your commitments
- They provide enablers for further progress
- They are only partially proven in DoD practice
  - Need further tailoring and piloting

Draft Conclusions

- Current SysE guidance much better than before
  - Still major shortfalls in integrating software, human factors
  - Especially with respect to future challenges
    - Emergent, rapidly changing requirements
    - High assurance of scalable performance and qualities
- ICM principles address challenges
  - Commitment and accountability, stakeholder satisficing, incremental growth, concurrent engineering, iterative development, risk-based activities and milestones
    - Can be applied to other process models as well
  - Assurance via evidence-based milestone commitment reviews, stabilized incremental builds with concurrent V&V
    - Evidence shortfalls treated as risks
  - Adaptability via concurrent agile team handling change traffic

Other (Ritter) Comments

- Other risks:
  - ability to do incremental
  - inability to articulate risks related to partners (not their output)
  - instability of multiple releases
- Risks in subprojects are not necc. project level risks
- If no HCI risks, then nothing needed from HCI

Common Risk-Driven Special Cases of the Incremental Commitment Model (ICM)

Special Case	Example	Size, Complexity	Change Rate % /Month	Criticality	NDI Support	Org, Personnel Capability	Key Stage I Activities : Incremental Definition	Key Stage II Activities: Incremental Development, Operations	Time per Build; per Increment
1. Use NDI	Small Accounting				Complete		Acquire NDI	Use NDI	
2. Agile	E-services	Low	1 – 30	Low-Med	Good; in place	Agile-ready Med-high	Skip Valuation , Architecting phases	Scrum plus agile methods of choice	<= 1 day; 2-6 weeks
3. Scrum of Scrums	Business data processing	Med	1 – 10	Med-High	Good; most in place	Agile-ready Med-high	Combine Valuation, Architecting phases. Complete NDI preparation	Architecture-based Scrum of Scrums	2-4 weeks; 2-6 months
4. SW embedded HW component	Multisensor control device	Low	0.3 – 1	Med-Very High	Good; In place	Experienced; med-high	Concurrent HW/SW engineering. CDR-level ICM DCR	IOC Development, LRIP, FRP. Concurrent Version N+1 engineering	SW: 1-5 days; Market-driven
5. Indivisible IOC	Complete vehicle platform	Med – High	0.3 – 1	High-Very High	Some in place	Experienced; med-high	Determine minimum-IOC likely, conservative cost. Add deferrable SW features as risk reserve	Drop deferrable features to meet conservative cost. Strong award fee for features not dropped	SW: 2-6 weeks; Platform: 6-18 months
6. NDI-Intensive	Supply Chain Management	Med – High	0.3 – 3	Med-Very High	NDI-driven architecture	NDI-experienced; Med-high	Thorough NDI-suite life cycle cost-benefit analysis, selection, concurrent requirements/ architecture definition	Pro-active NDI evolution influencing, NDI upgrade synchronization	SW: 1-4 weeks; System: 6-18 months
7. Hybrid agile / plan-driven system	C4ISR	Med – Very High	Mixed parts: 1 – 10	Mixed parts; Med-Very High	Mixed parts	Mixed parts	Full ICM; encapsulated agile in high change, low-medium criticality parts (Often HMI, external interfaces)	Full ICM ,three-team incremental development, concurrent V&V, next-increment rebaselining	1-2 months; 9-18 months
8. Multi-owner system of systems	Net-centric military operations	Very High	Mixed parts: 1 – 10	Very High	Many NDIs; some in place	Related experience, med-high	Full ICM; extensive multi-owner team building, negotiation	Full ICM; large ongoing system/software engineering effort	2-4 months; 18-24 months
9. Family of systems	Medical Device Product Line	Med – Very High	1 – 3	Med – Very High	Some in place	Related experience, med – high	Full ICM; Full stakeholder participation in product line scoping. Strong business case	Full ICM. Extra resources for first system, version control, multi-stakeholder support	1-2 months; 9-18 months

C4ISR: Command, Control, Computing, Communications, Intelligence, Surveillance, Reconnaissance. **CDR:** Critical Design Review.
DCR: Development Commitment Review. **FRP:** Full-Rate Production. **HMI:** Human-Machine Interface. **HW:** Hard ware.
IOC: Initial Operational Capability. **LRIP:** Low-Rate Initial Production. **NDI:** Non-Development Item. **SW:** Software

Where does this leave us in HCI?

(Pew & Mavor, 2007, ch. 3)

All HCI techniques can be seen as a way to reduce risk

- Define opportunities and context of use:
  scenarios, personas, task analysis
- Define requirements and design solutions:
  TA, models
- Evaluate:
  VPA, behavior loggers (e.g., RUI)

Shared Representations as Part of Design Process - Uses

- Examined critically
- Reduce working memory load
- Make explicit what is explicit and implicit
- Produce new connections
- Collaboratively produce new knowledge
- *Transfer knowledge*

Shared representations - Attributes

- Help establish a shared representation
- Facilitate desired social processes (*and cognitive processes*)
- Provide strategically chosen ambiguity
- Make differences and relationships apparent
- Facilitate ‘group thinking’
- Provide meaningful structure, content, and appearance to creators and consumers

APPENDIX 3-A

TABLE 3-A1 Best Practices for Risk Mitigation

Activity Category	Best Practices for Risk Mitigation from ISO/PAS 18152	Example HSI Methods and Techniques
1. Envisioning opportunities	–Identify expected context of use of systems [forthcoming needs, trends and expectations]. –Analyze the system concept [to clarify objectives, their viability and risks].	–Field observations and ethnography –Participatory analysis
2. System scoping	–Describe the objectives which the user or user organization wants to achieve through use of the system. –Define the scope of the context of use for the system.	–Organizational and environmental context analysis –Field observations and ethnography –Participatory analysis *Work context analysis*
. . .		
5. Architecting solutions (a) System architecting	–Generate design options for each aspect of the system related to its use and its effect on stakeholders. –Produce user-centered solutions for each design option. –Design for customization. –Develop simulation or trial implementation of key aspects of the system for the purposes of testing with users. –Distribute functions between the human, machine, and organizational elements of the system best able to	–Task analysis –Work domain analysis –Participatory design –Prototyping –Models and simulations *Function allocation* *Generate design options*

. . . .

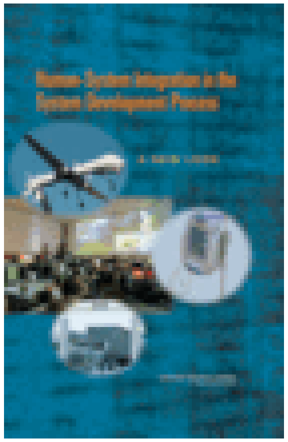
References

Boehm, B., & Hansen, W. (2001). The Spiral Model as a tool for evolutionary acquisition. *Crosstalk: The Journal of Defense Software Engineering*, May. 4-11.

Boehm, B. (2007). Integrating Hardware, Software, and Human Factors into Systems Engineering via the Incremental Commitment Model. Stevens Presentation.

Booher, H. R., & Minninger, J. (2003). Human systems integration in Army systems acquisition. In H. R. Booher (Ed.), *Handbook of human systems integration* (pp. 663-698). Hoboken, NJ: John Wiley.

HIS in the SDP Chapter 4: Managing Risks



Frank Ritter, with help from Barry Boehm

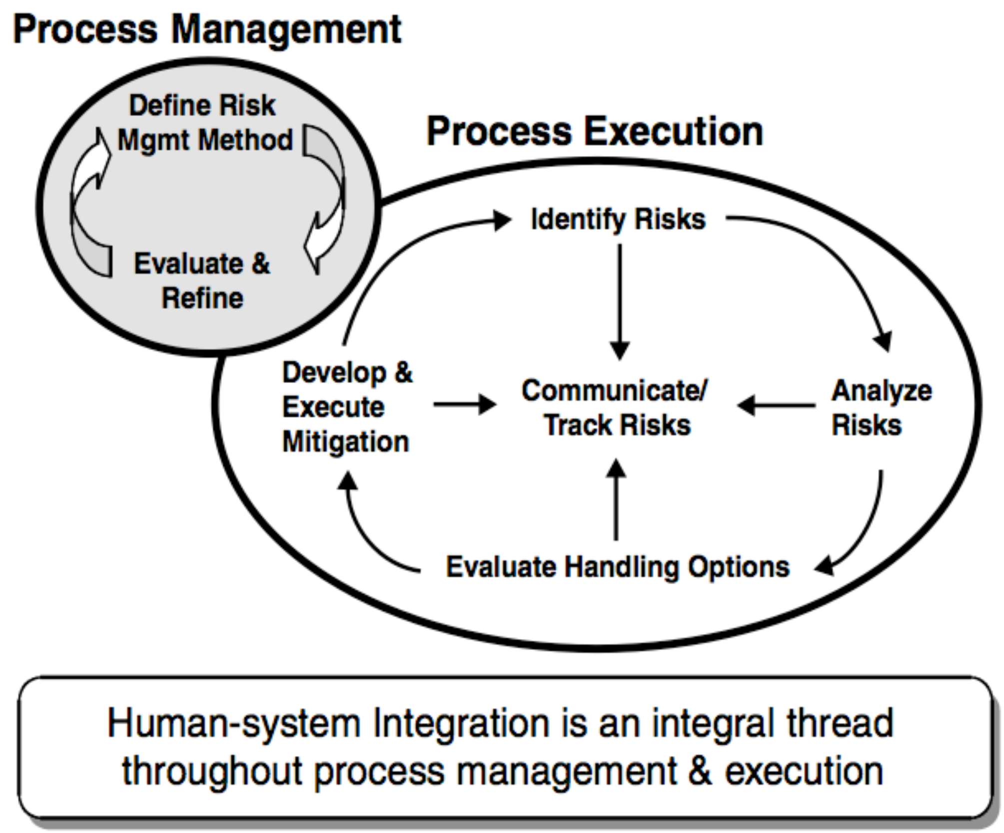
29 jan 08

Glossary

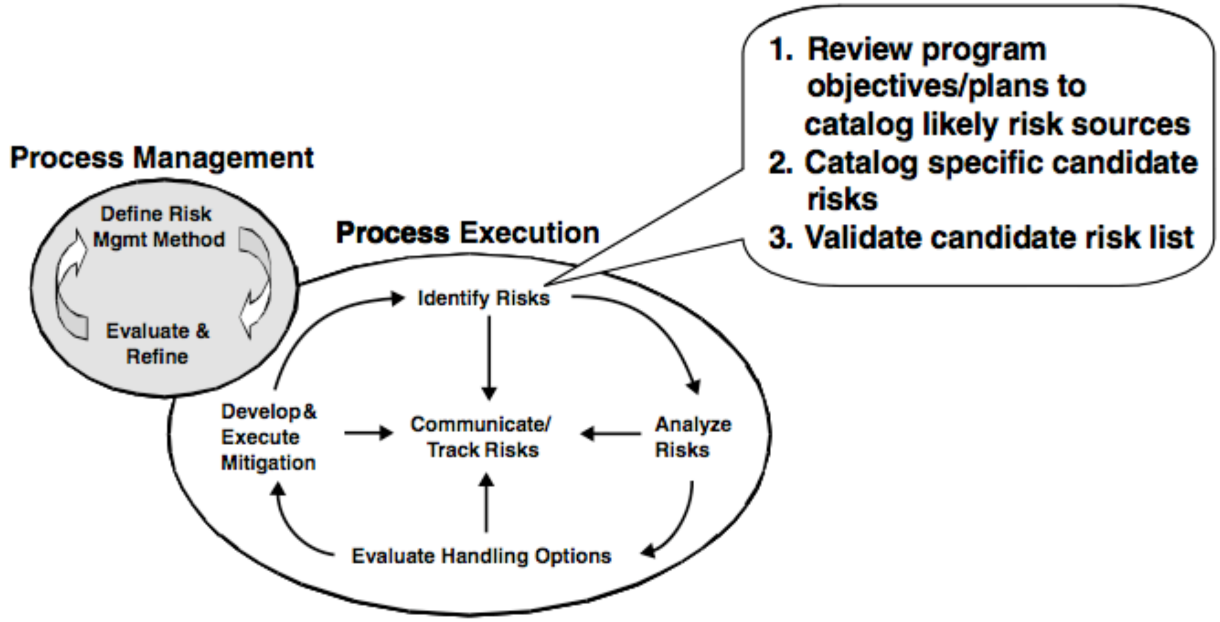
- ACR architecting commitment review
- COTS commercial off the shelf
- ICM incremental commitment model
- LCA life cycle architecture
- OODA loop by Boyd

The Risk Management Process

- Good practices for program management
  - Assumes a stakeholder analysis (e.g., business offer, proposal, specification)
  - Including HSI in this process
  - A program organization
  - Culture of openness

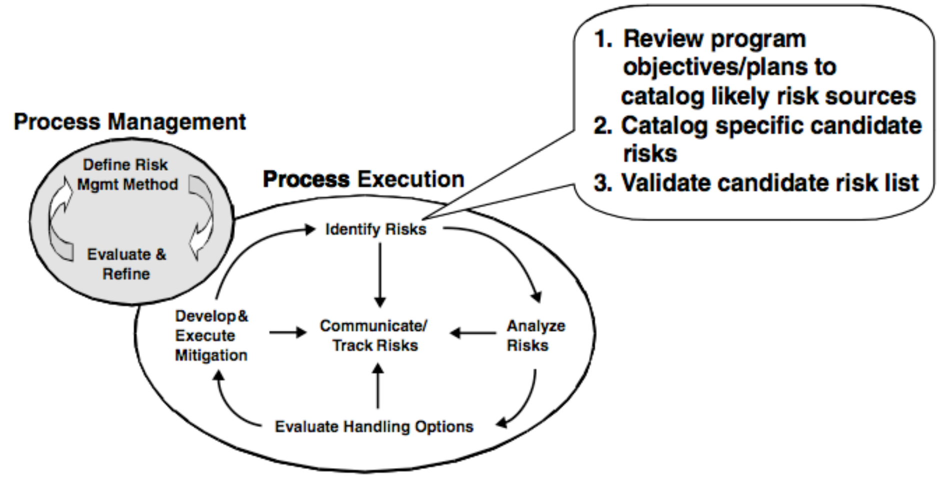


The Risk Management Process: Identification



- Risk identification not formalized, but
- Involve all stakeholders (e.g., users! Developers, HIS, trainers)
- Iterate risk identification until program completed
- Use nonadvocate technical experts to assist with risk identification
- Encourage identification and recording
- Set up process to afford consistent documentation

The Risk Management Process: Identification

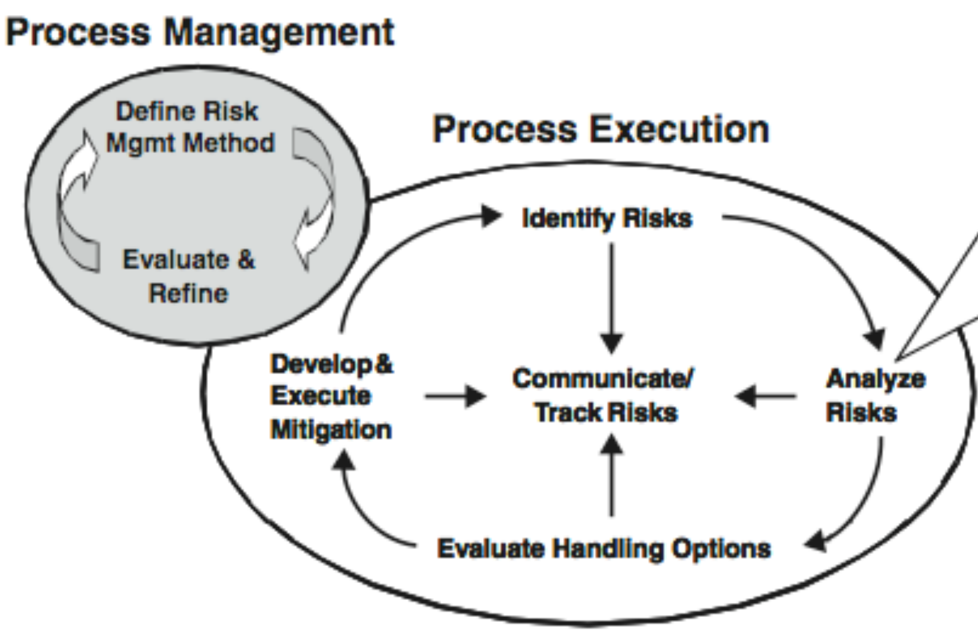


Example risks:

- Performance does not satisfy user requirements
- Performance does not match other stake holder requirements
- Mismatch of system to context (sand in tools)
- Ability to incorporate HSI to reduce risks including Wrong types of developers and HIS professionals

- Also see Booher and Minniger for long lists of risks that were realised, CMU tech report in Boehm and Hansen, and London ambulance disaster from *Set phasers on stun*
- People see what they are trained to see

The Risk Management Process: Analyse

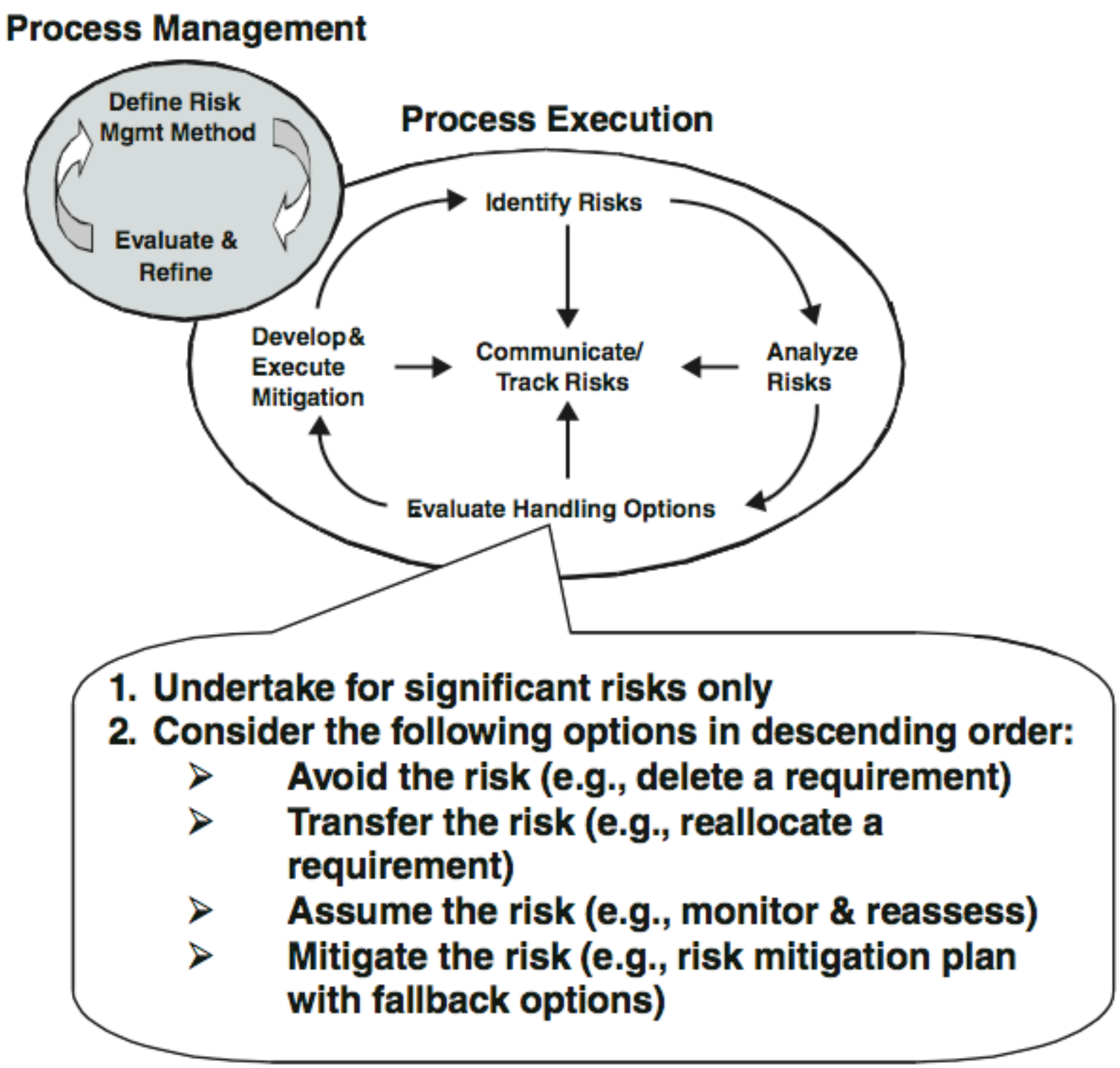


1. Define analysis method
2. For candidate risks & executed risk mitigation plans, determine:
   - Likelihood & consequence
   - Risk level
3. Assess impacts to program
4. Prioritize & denote significant risks

<u>Comments</u>:

- Communication of risks matters (see Rosling talk on the developing world, www.gapminder.org/video/talks/)

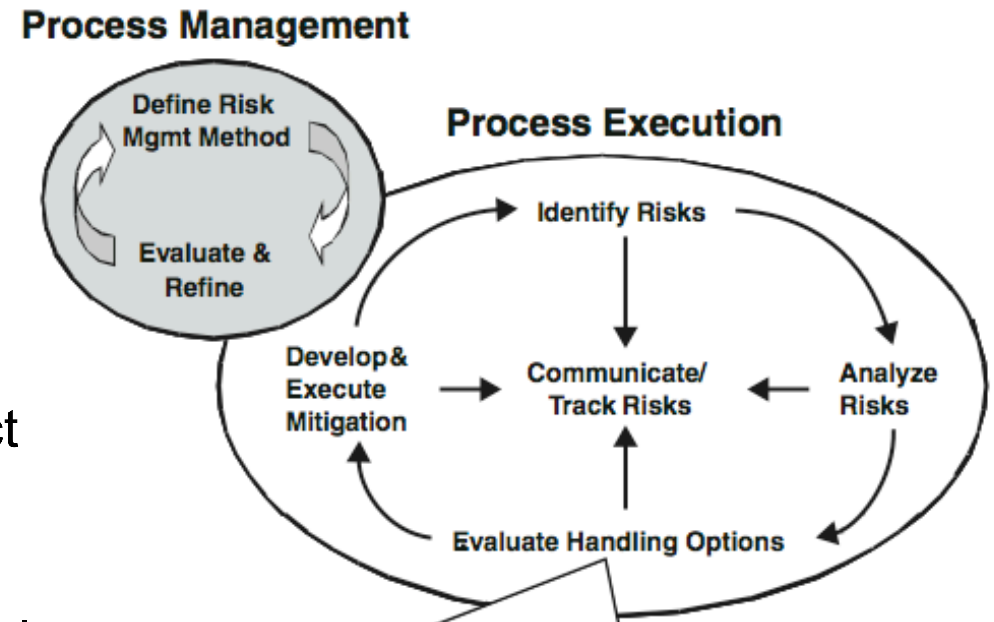
The Risk Management Process: Handling Options



<u>Comments</u>:

- Dealing with large risks
- HSI has a set of tools for these options, more for avoid (know user and task), Assume (monitor), Mitigate (understand, modify)

The Risk Management Process: Execute Handling



1. Undertake for significant risks only
2. Consider the following options in descending order:
   - Avoid the risk (e.g., delete a requirement)
   - Transfer the risk (e.g., reallocate a requirement)
   - Assume the risk (e.g., monitor & reassess)
   - Mitigate the risk (e.g., risk mitigation plan with fallback options)

<u>Comments:</u>

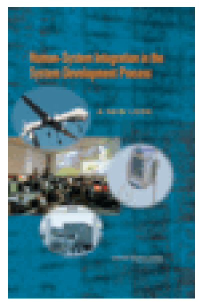
- These risks may interact
- Need to be monitored
- New ones will arise
- Need to be part of formal process, else, problems can occur
- Ritter’s impression is that in normal progress, risk sizes decrease over time

References

Coram, R. (2002). *Boyd: The fighter pilot who changed the art of war*. New York, NY: Back Bay Books/Little, Brown and Company.

Ritter, F. E. (2005). Review of "Boyd: The fighter pilot who changed the art of war". *The Military Psychologist, The Official Newsletter of Division 19 of the APA, 21*(2), 21. www.apa.org/divisions/div19/.
(Similar loop, perhaps, but different domain)

Part II: HSI Methods in system development

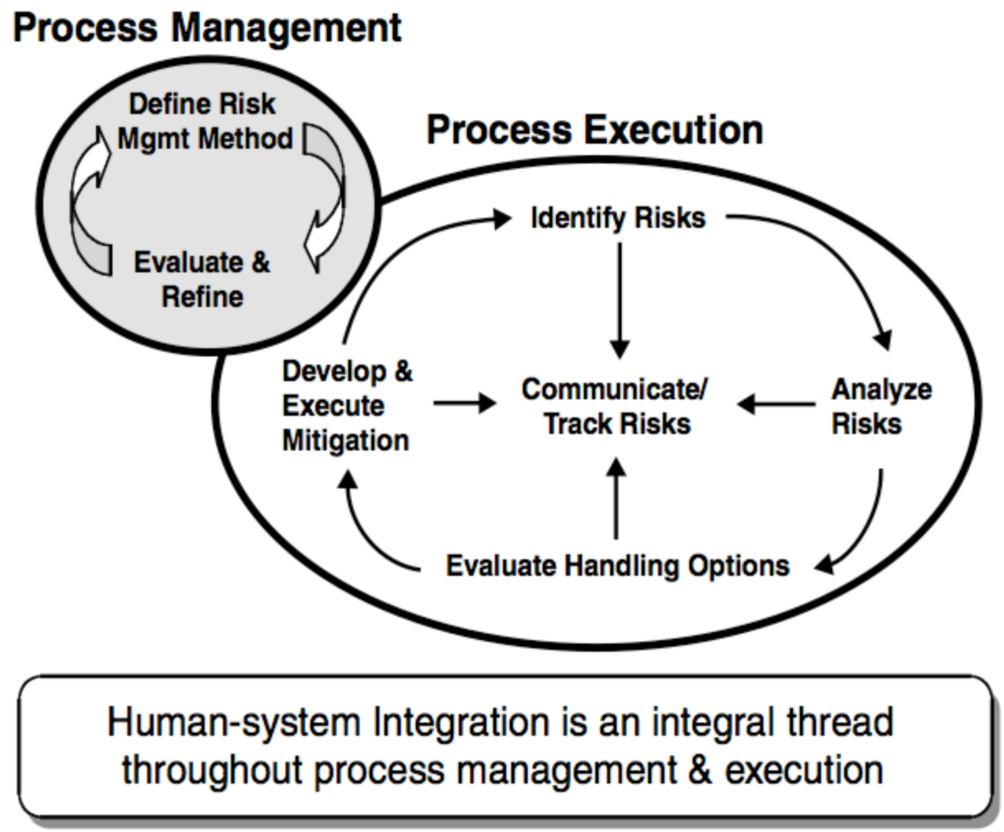


Frank Ritter, without help from Barry Boehm this time

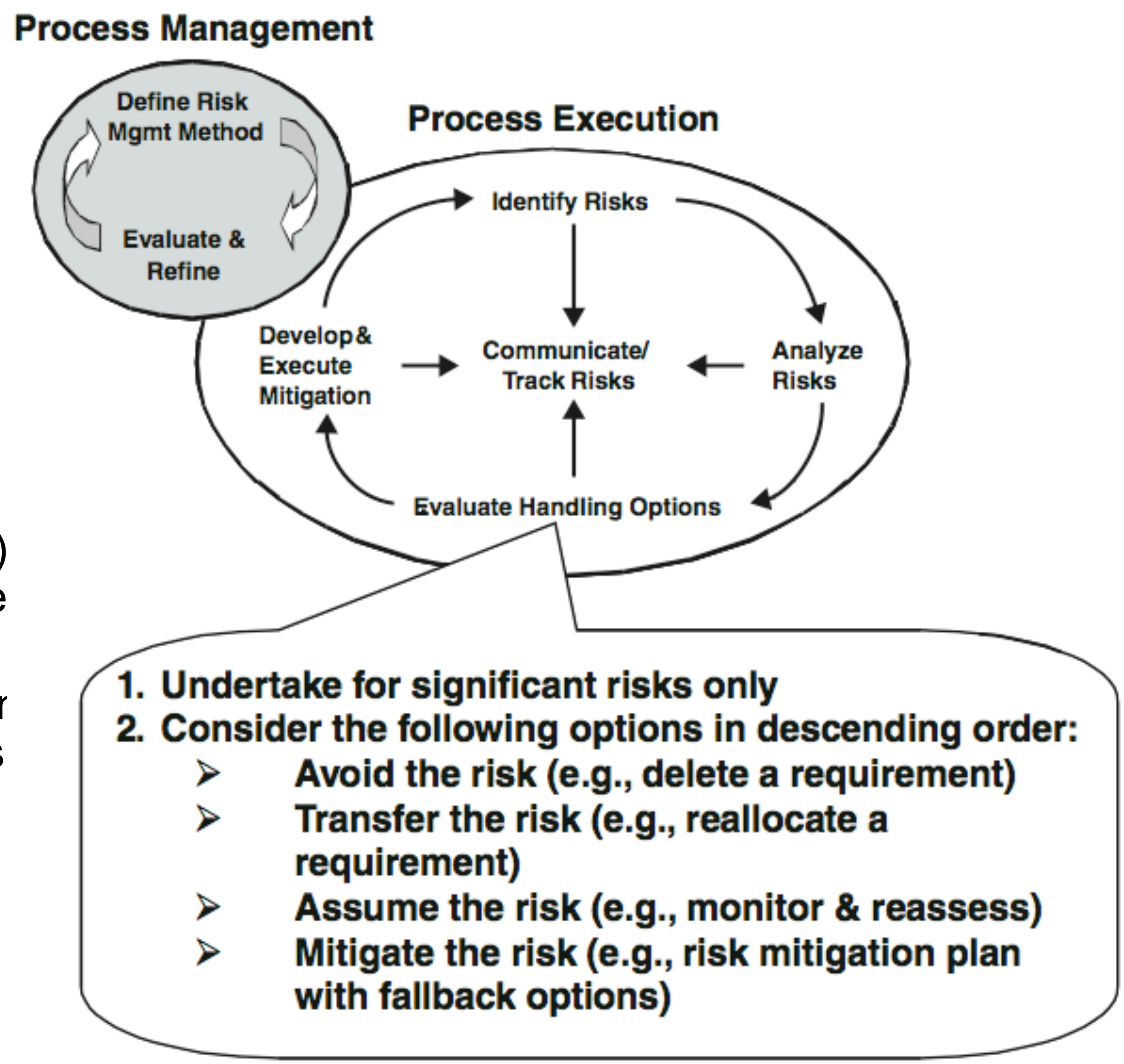
5 feb 08

The Risk Management Process

- Good practices for program management
  - Assumes a stakeholder analysis (e.g., business offer, proposal, specification)
  - Including HSI in this process
  - A program organization
  - Culture of openness



The Risk Management Process: Handling Options



Comments:

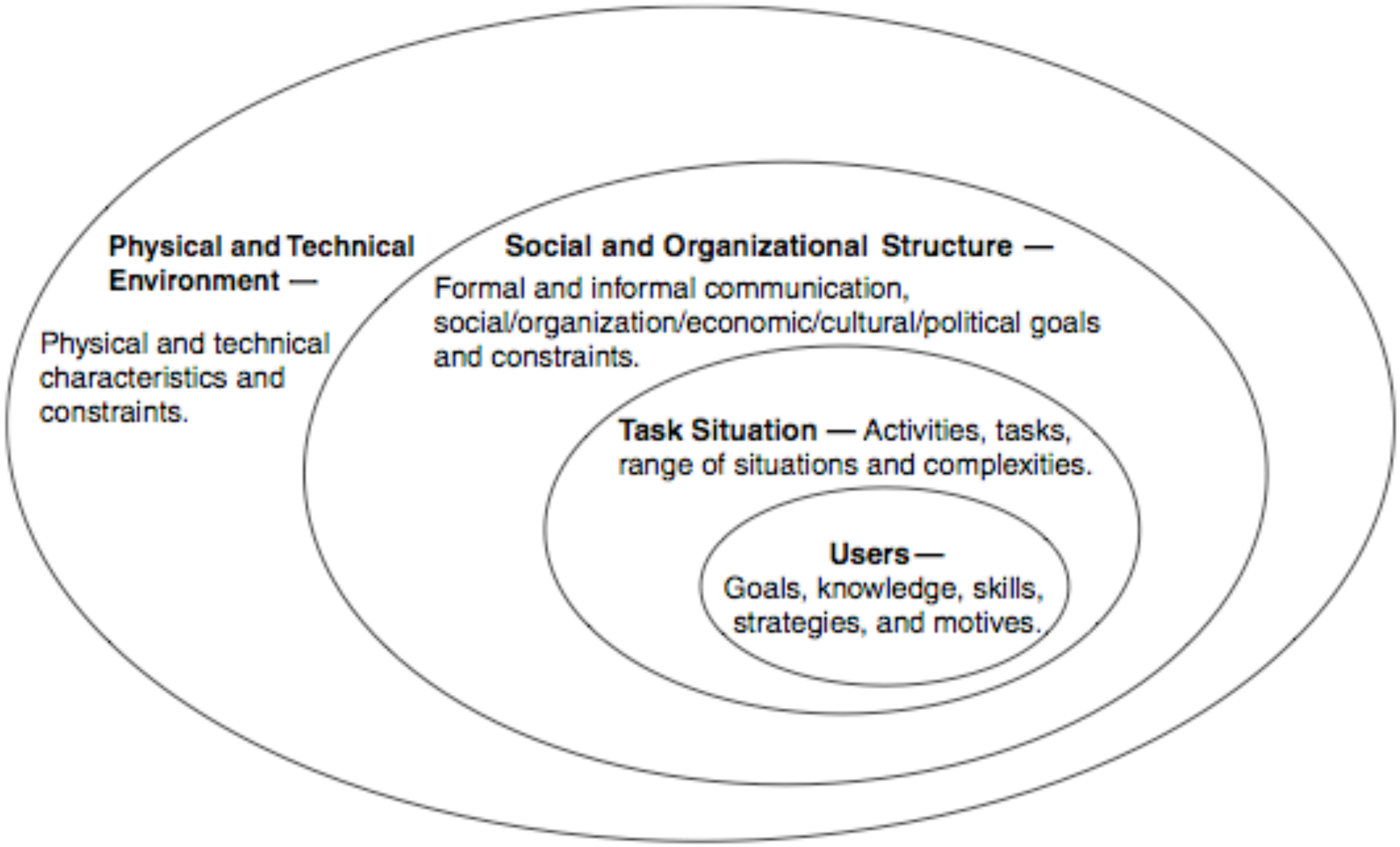
- Dealing with large risks
- HSI has a set of tools for these options, more for avoid (know user and task) Assume (monitor), Mitigate (understand, modify)
- Ritter's impression is that ir normal progress, risk sizes decrease over time

Methods

- Three major periods of use
  - Define context of use
  - Define requirements and design solutions
  - Evaluate

- All fit back into spiral, all used to reduce risks using previous approaches
- We have bags of these methods!
- Classification to period is somewhat arbitrary
- Not exhaustive, *illlustrative*
- Function allocation not covered
- Performance measurement details not covered

Area 1: Context of Use



- Helps avoid local optimizations, feature creep, unanticipated effects

TABLE 6-4 Examples of Uses of Event Data Analysis

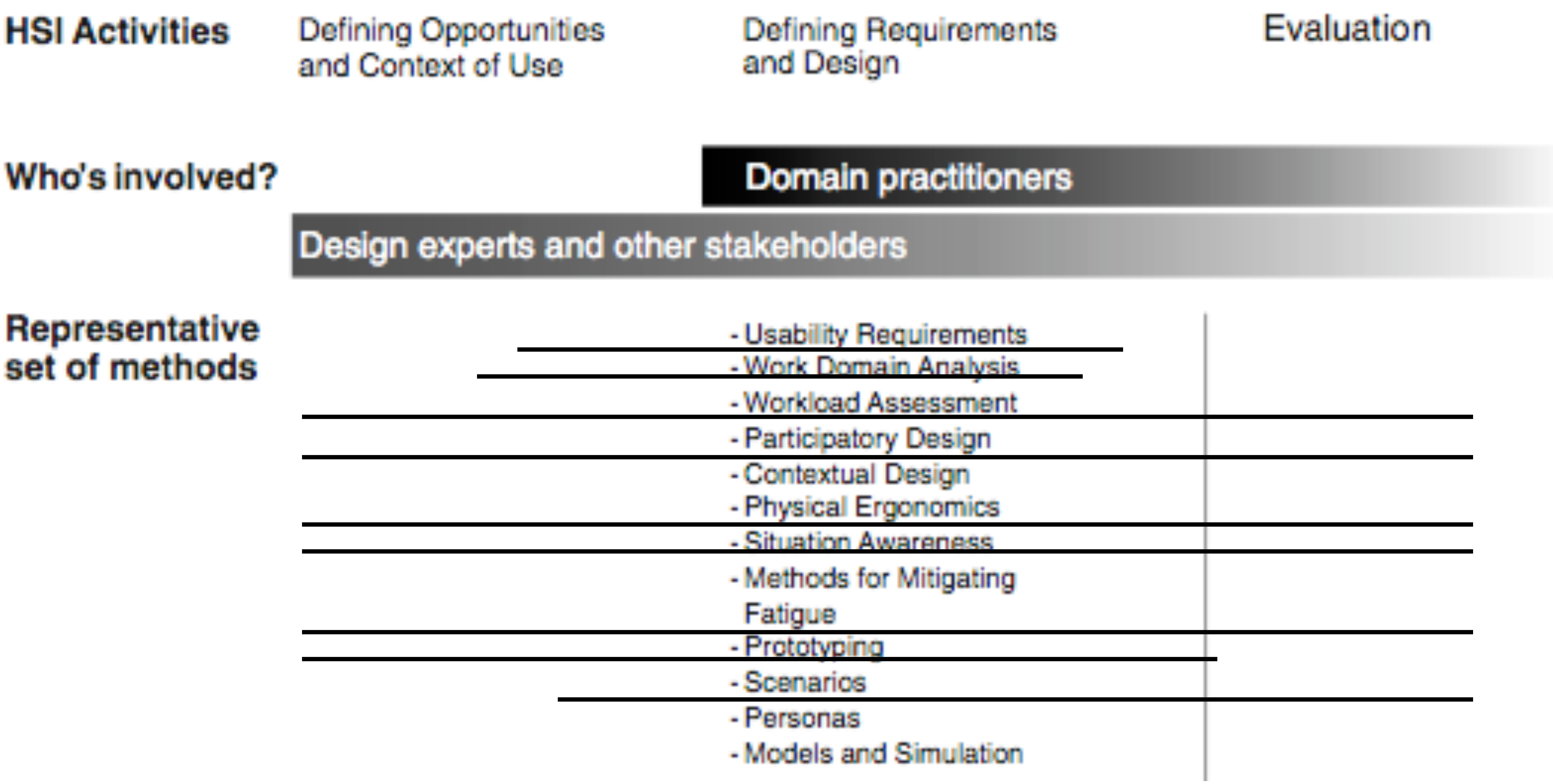
Question	Type of Event Data
What does the operator do from moment to moment? What options are not used? What options precede the request for help? What action sequences occur often enough to be automated or assisted?	Keystrokes, mouse movements, click streams.
What are the service demands made on a shared resource (like a server or a database)? What are critical dates or times of day? How can server/database traffic be anticipated or smoothed?	Hits on a web site. Database accesses. Server traffic. (While conventional server logs provide a very low-level view of these demands, instrumentation can provide a work-oriented account of server demands.)
What are the current issues that the organization is grappling with? What is the organization's current intellectual capital?	User-initiated social-software events and data, like tag creation and tag modification, blog entries, wiki entries, and current searches.
What are people thinking and planning as they work? What confuses them?	Think-aloud reports. Verbal reports. Paired-user testing.
What is the communication network in the organization? Who communicates with whom?	Communications events (email, chat, meeting attendance).
What is the context of critical events? How often do critical events occur and what events preceded and follow them?	Stream of video events (e.g., in an emergency room or air traffic control center). One or more recordings of shared radio frequencies

TABLE 6-5 Life-Cycle Phases of the ICM and EDA

Phase	Method	Variation
Exploration	EDA	May help scope problem; can base on expert judgment if no existing system.
Valuation	EDA	Use to describe existing behavior; highlight obvious weaknesses, strengths.
Architecting	EDA	Begin to focus more on future behavioral repertoire; change to existing behavior patterns.
Development	EDA-E	Can collect behavioral data with prototype and evaluate success of new design.
Operation	EDA-E	Given other criterion can collect data from users in beta testing to assess success.

NOTE: EDA-E (Evaluative) includes evaluative steps such as assessment and diagnosis.

Stretch of these tools



Usability requirements

- Usability is not likability (seen in Rossen and Carroll chapter)
- Hard to know if systems will meet these measures
- Don’t have good measures and standards
- Optimizes what is measured

Models

- Risk: we are not like we think we are
- Running models in our head is hard particularly without a PhD in pschology
- But models hard to use
- But but working on models to be more usable
- Insight: perhaps especially here, designers learn for the next design

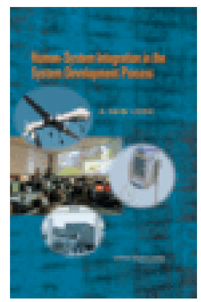
Area 3: Methods for Evaluation

- Also see all previous methods

- Risk Analysis
- Usability Evaluation Methods
- User satisfaction ratings
- Surveys/questionnaires
- Interviews
- Experiment design
- Statistics
- Performance measurement

- Failure Modes and Effects Analyses (FMEA)
- Fault Tree Analyses (FTA) and other technique variations
- Lists of usability problems in the form of: written reports, presentations, or videos
- Time and accuracy of user's performance

Part III: The Future: Scenarios, Conclusions, and Recommendations
[of HSI Methods in System Development]



Frank E. Ritter

12 feb 08

(presented 19 feb 08)

Review of So Far

- Risk driven
- Incrementally growing
- Basis for agreement among stakeholders
- Covered methods, tools, and shared representations

  Noted gaps, and needed methodologies and tools are in book

Future Scenarios

- 5-10 years [!-fer]

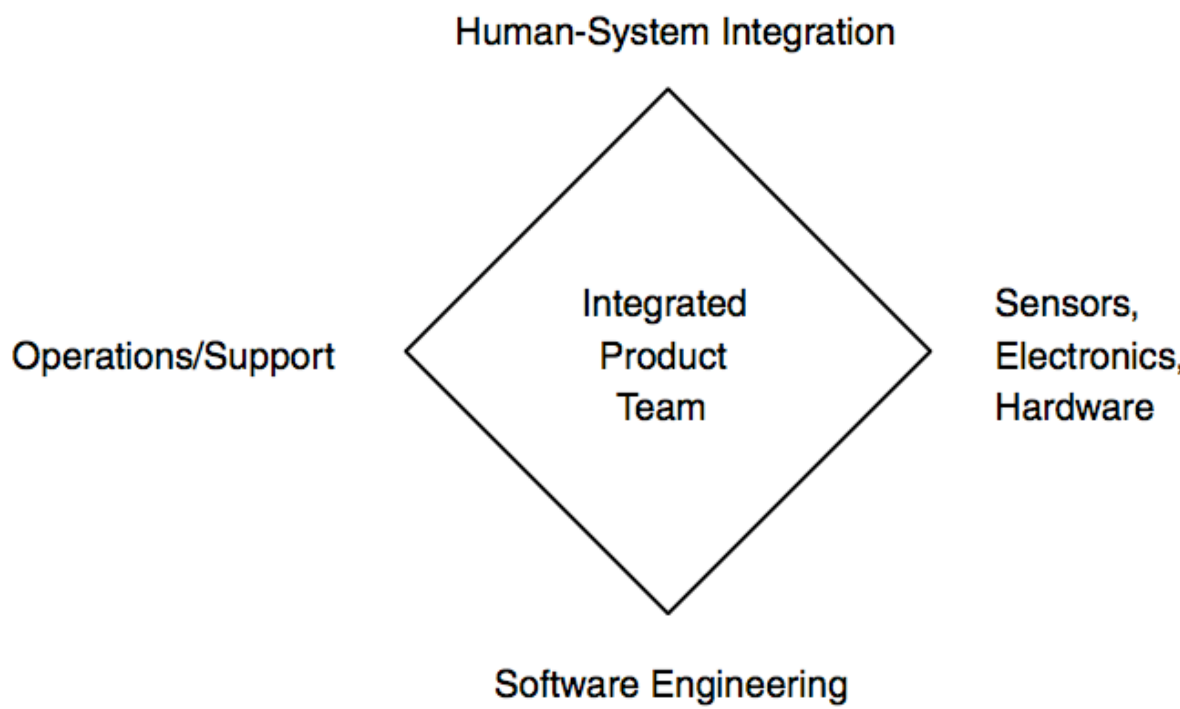
1. Integrated methodology
2. Developing HSI as a discipline
3. Knowledge-based planning
4. (Greater) User participation

1. An Integrated Methodology

- Generate a quantitative baseline
- Define opportunities and requirements, and context of use
  - Broad use of Shared Representations
- Design solutions
  - Priorities based on risks
  - Shared representations developed, e.g.,
    - From personas to running models
    - Gantt charts become time-based and synched with scenarios and prototypes
    - Scripted modules to hardware and software
    - Software from designs to code (seamlessly (!))
- Evaluation
  - Including model-based and stakeholder evaluation at the end
- Integration thus means:
  - Across stages of shared representations
  - Builds upon previous stages results
  - Teams integrated across stages
  - System integrated before release

1. An Integrated Methodology

- HSI-led teams
- To avoid risks to mission, risks to usability
  - Booher & Minneger, 2003 have numerous examples
- Use of integrated product teams (IPT) (Rouse, 2005)



2. Developing HSI as a Discipline

- Related disciplines
  - Experimental psychology
  - Industrial engineering
  - Information sciences and technical writing
  - Traditional systems engineering
- Workshops and continuing ed. programs
- The use of prakticums
- HSI tracks at conferences and in journals
- (Tutorials)

3. Knowledge-based Planning

- Tools to help acquire system-specific knowledge related to risks
- Inputs
  - Size, organizational complexity, precedents, criticality, available expertise
- Outputs
  - Summary of risks to be managed
  - Development timelines and staffing profiles
  - Most relevant tools and methods

4. Greater User Participation

- Context of use methods can be expensive
- Approaches to capturing user input (and creating mods)
  - Combine lists with maps (mash-ups)
  - RSS feeds and associated tools
  - Social bookmarks
  - Blogs and associated multimedia
  - Wikis
- Systems Engineering for User Participation in these approaches
  - Building tools and systems to support users in this process
  - Design for end user customization
  - Support issue tracking and resolution

Conclusions

- Include HSI early, understand how to do it
- Tailor methods to risk and resources
- Ensure communication of shared representations (models of various things)
- Design to accommodate change
- Projects to develop process
- Projects to implement HSI as a field
- Projects to improve models (ease to create, ease to understand, quality), shared representations, data analysis
- Projects to improve usability objectives

Further Insights

- Insight: Impact on next project
  - Size of users tasks, complexity of tasks, their interrelation, scope
  - May be true for all these methods
  - So shared to next design, and understanding of designer
- Insight: They think they do it already
  - Good, buy in to part
  - Bad, already know how
  - Insight: need to give counter examples
- Insight: Education and sharable representations are more important than one might think

Thank you, the end.