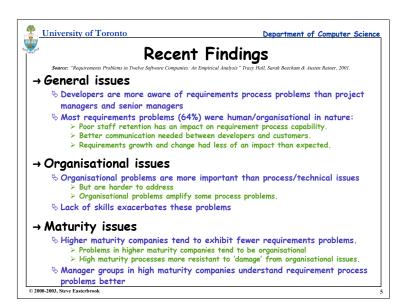
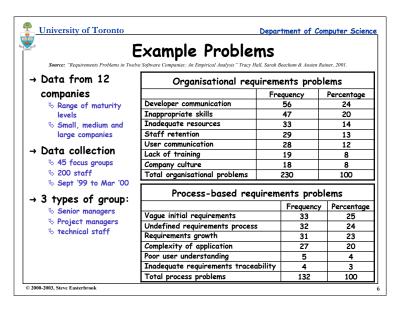
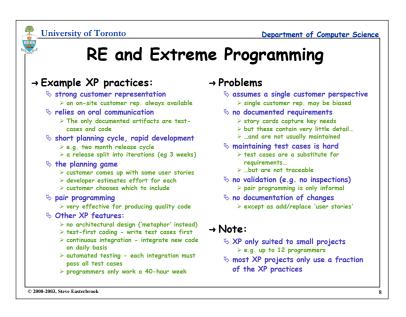


Capability Maturity Model Source: Adapted from Humphrey, 1989, chapter 1				
Level	Characteristic	Key Challenges		
5. Optimizing	Improvement fed back into process	Identify process indicators "Empower" individuals		
4. Managed	(Quantitative) measured process	Automatic collection of process data Use process data to analyze and modify the process		
3. Defined	(Qualitative) process defined and institutionalized	Process measurement Process analysis Quantitative Quality Plans		
2. Repeatable	(Intuitive) process dependent on individuals	Establish a process group Identify a process architecture Introduce SE methods and tools		
1. Initial	Ad hoc / Chaotic No cost estimation, planning, management.	Project Management Project Planning Configuration Mgmnt, Change Control Software Quality Assurance		



Barriers to cultural change			
Obstacles	Examples	But experience shows	
"It is not worth discovering needs directly from users"	"We've been developing such products for a long time and know users' needs"	Studies show developers are often surprised by user behaviour and expectations	
	"We also use our own products and can act as users ourselves"	Developers tend to be biased by their technical expertise	
	"It's a new product - therefore users cannot have any needs for it"	Still need to understand the current context and existing tasks	
"It is difficult to discover needs directly from users"	"Users are unable to say what they need and want"	Combination of observation and other elicitation techniques works	
	"There are so many users we cannot interview them all"	It is possible and useful to identify representative potential users	
"It is risky to discover needs directly from users"	"Customers might think we don't know the basics of their business"	A well planned site visit improves the	
	"May spoil relations with the customer by asking stupid questions"	developer's image among customers	
"It is not worth documenting user requirements systematically"	"Customers want to see the technical specs, not user regts"	investigating needs often reveals that a technical solution won't work	
	"Documenting the requirements takes too much time"	Documented requirements save time later	







Department of Computer Science

Course Summary

→ Course Goals

- SExamine the state-of-the-art for research & practice in RE.
 - > Role of RE in software & systems engineering
 - > Techniques, notations, methods, processes & tools used in RE
- ♥ Gain practical experience in selected RE techniques
- ♥ Understand the essential nature of RE
 - > Breadth of skills needed for RE. and disciplines on which it draws > Contextual factors & practicalities
- ♥ Gain a basic grounding for research in RE
 - > Methodological issues for research
 - > Current research issues & direction of the field
 - > Awareness of the literature

→ Course Syllabus

- ♥ Introductory stuff > What is RE?
 - > Why is it important?
- **♥** Foundations
- > inter-disciplinary aspects of RE
- **♥** Basic RE activities
- > Eliciting Requirements
- > Modelling & Analysing Requirements
- > Communicating Requirements > Agreeing Requirements
- > Evolving Requirements
- ♥ Integrated RE
 - > Method Engineering
 - > Patterns and Problem Frames

© 2000-2003, Steve Easterbrook



University of Toronto

Department of Computer Science

Feedback Questions

Did the course meet your expectations? How useful do you think the course was to you? What do you feel you have learned? What did you not learn, that you had hoped to? What was the best part of the course? What was the worst part of the course? How might the course be improved in the future?

© 2000-2003, Steve Easterbrook