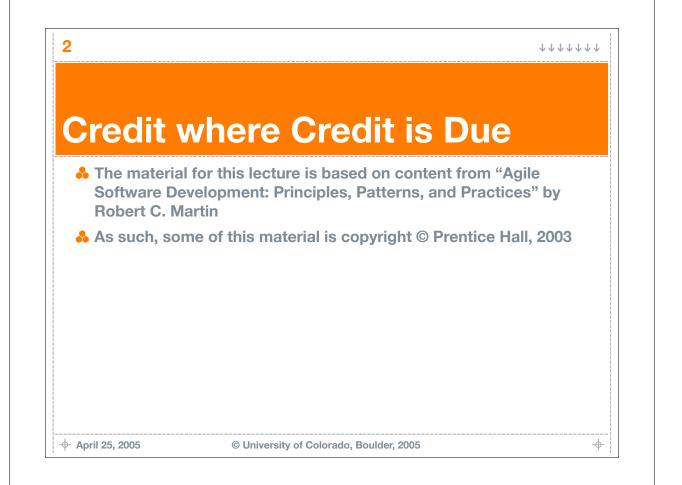
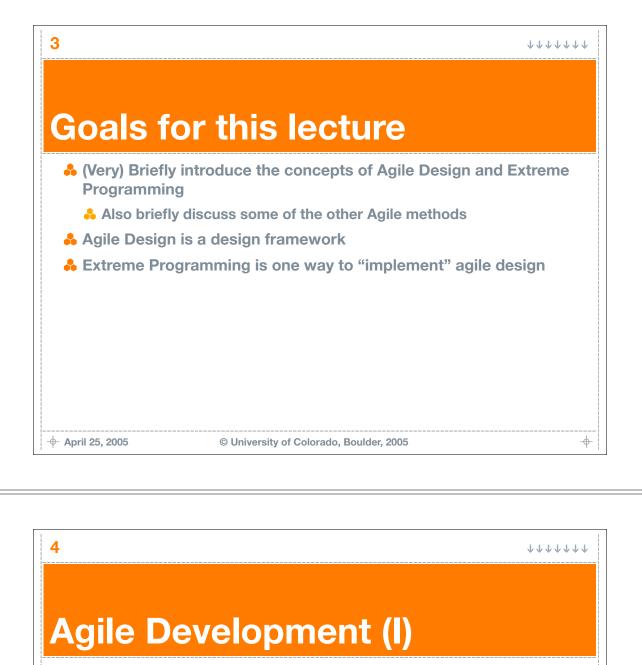
Lecture 29: Agile Des	ign and
Extreme Programmin	
Extreme Programmin Kenneth M. Anderson	
Lecture 29: Agile Des Extreme Programmin Kenneth M. Anderson Software Methods and Tools CSCI 4448/6448 - Spring Semester, 2005	

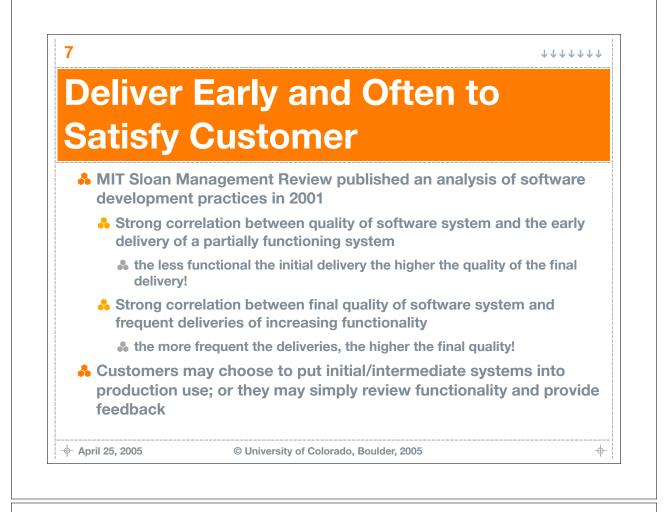


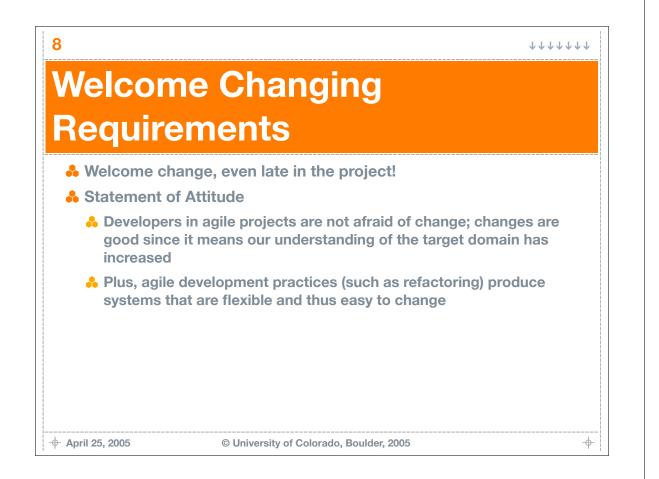


- Agile development is a response to the problems of traditional "heavyweight" software development processes
 - too many artifacts
 - too much documentation
 - 🔒 inflexible plans
 - Iate, over budget, and buggy software

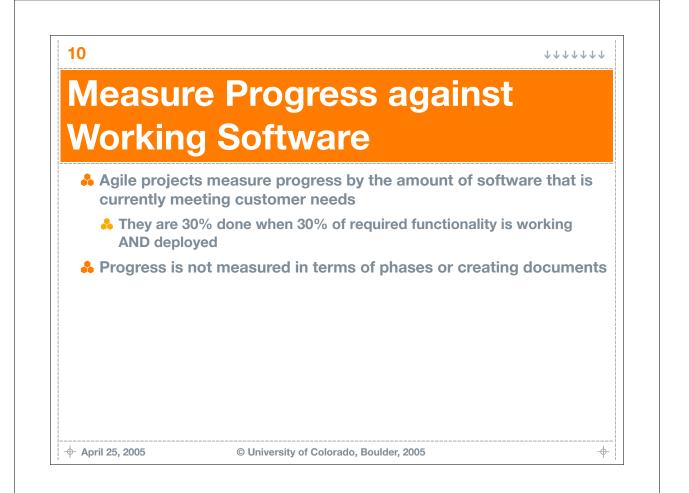
gile	Development (II)
A manife	sto (from the Agile Alliance)
	e uncovering better ways of developing software by doing it Iping others do it. Through this work we have come to value
🖧 indiv	iduals and interactions over processes and tools
🌲 worl	king software over comprehensive documentation
🔥 cust	omer collaboration over contract negotiation
👪 resp	onding to change over following a plan
	, while there is value in the items on the right, we value the on the left more



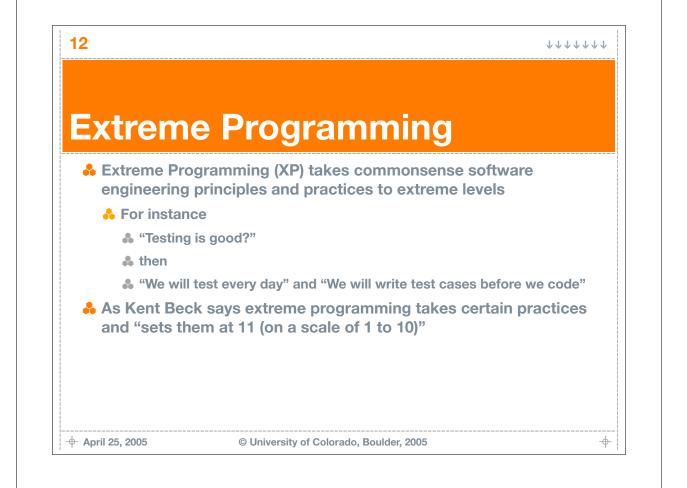




Best		
🔒 In an agile	e project, people talk to each other!	
👶 The prin	nary mode of communication is conversation	
there	is no attempt to capture all project information in writing	
	s are still created but only if there is an immediate and ant need that they satisfy	
🎄 they r	nay be discarded, after the need has passed	



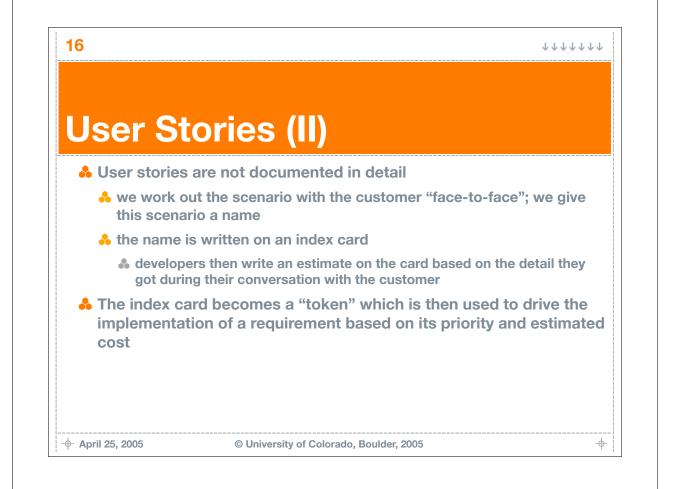
This refers	to the art of maximizing the amount of work NOT done
Agile pro 👶	jects always take the simplest path consistent with their joals
🔥 They do today's p	not try to anticipate tomorrow's problems; they only solve problems
• •	ality work today should provide a simple and flexible system be easy to change tomorrow if the need arises

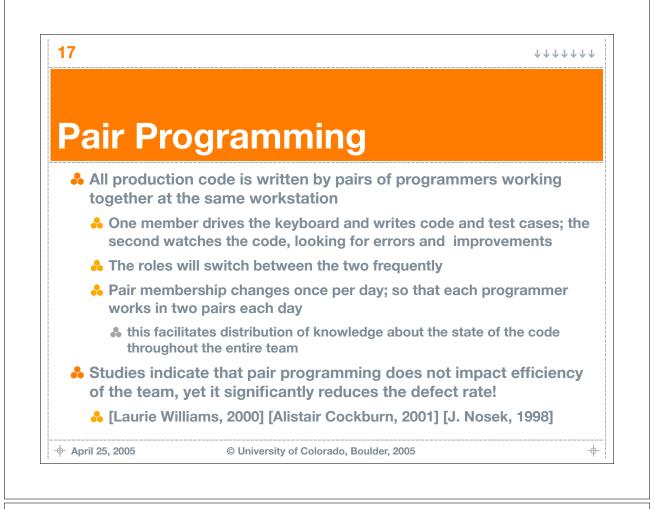


🔥 Th	e best way to describe XP is by looking at some of its practices
	There are fourteen standard practices, we'll look at six important ones
	Customer Team Member
	& User Stories
	Pair Programming
	Test-Driven Development
	Sollective Ownership
	& Continuous Integration



	Stories (I)	
	to have requirements	
🔥 We ne	irements come in the form of "user stories" or scenarios eed just enough detail to estimate how long it might take to op software to support this story	
star	id too much detail, since the requirement will most likely change; rt at a high level, deliver working functionality and iterate based on licit feedback	



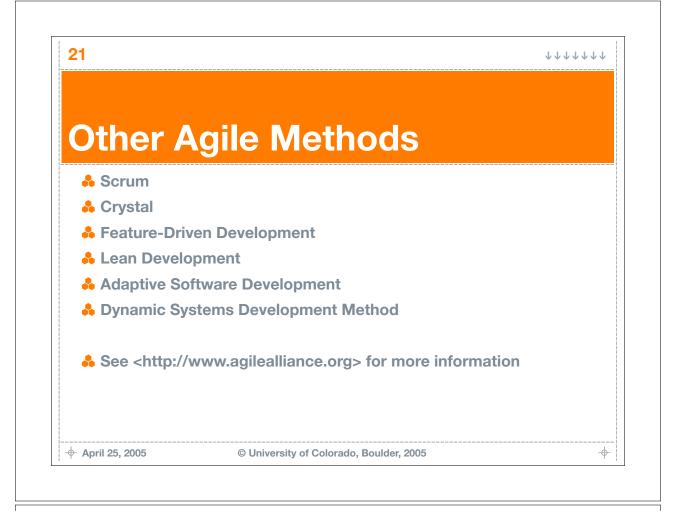




	ive Ownership	prove it
Develope or techno	rs are never individually responsible for a par logy	ticular module
🔒 This conce	ot is a hard one for people to adopt, you w	vill often hear

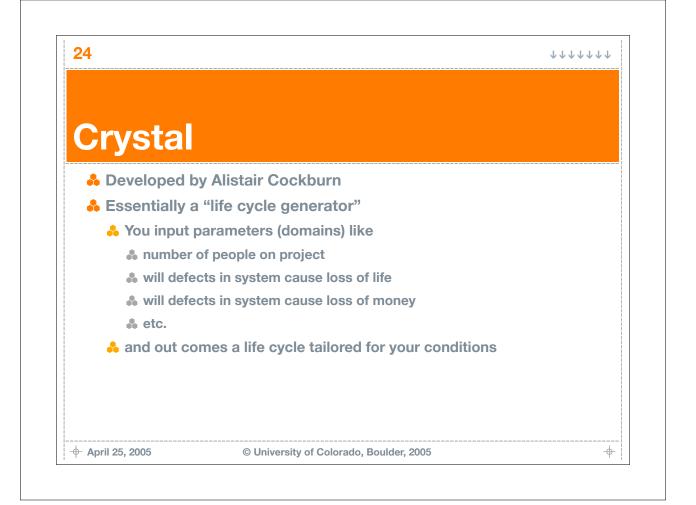
Continuous Integration

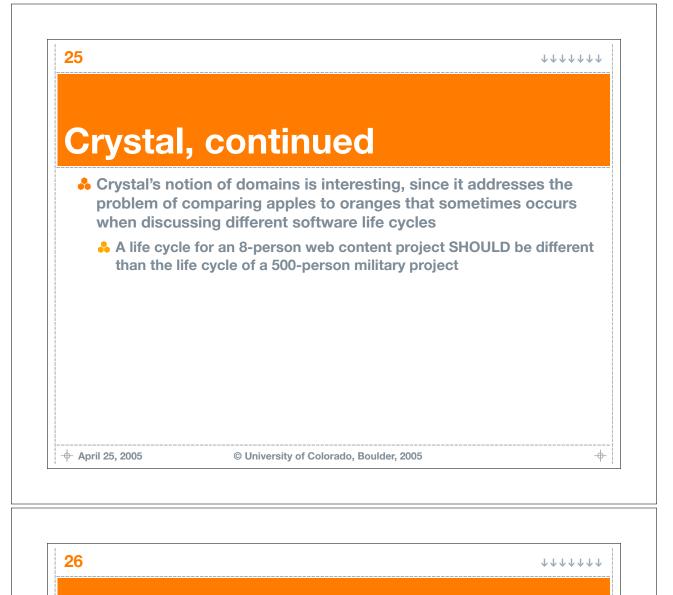
- Developers check in code and integrate it into the larger system several times a day
- Simple Rule: first one to check-in "wins"; everyone else merges
- Entire system is built every day; if the final result of a system is a CD, a CD is burned every day; if the final result is a web site, they deploy the web site on a test server, etc.
 - This avoids the problem of cutting integration testing to "save time and money"





Scrum.	continued	
	Continuou	
🔒 Involves two	lists	
🔒 Product Ba	acklog	
🔥 essentia	ly features of the desired system	
🔥 Scrum Bac	klog	
features	for the next "scrum"	
🔒 Involves thre	e phases	
🔥 Pre-Scrum	planning meeting	
🔥 Scrum		
🔒 Post-Scrur	n demo and debriefing	
-	ou are going to do, do it, and then demo t our clients and get ready for the next plan	
→ April 25, 2005	© University of Colorado, Boulder, 2005	





Feature Driven Development

- Developed by Jeff De Luca and Peter Coad
- Simple life cycle
 - Develop an overall system model
 - 👶 Build a feature list
 - Plan by feature
 - 🔒 Iterate
 - Design by feature
 - **Build by feature**

+ April 25, 2005

DD,	continued
🔒 Has one	notable success story
🔒 "The S	ngapore Project"
	ms integration firm had failed to produce a system to ing commercial loans for a large Singapore bank
👪 Afte	two years, they gave up, claiming the project was undoable
	d produced 3500 pages of use cases, an object model with hundred asses, thousands of attributes, but no methods and no code!

