## Software Development Challenges

#### Hard to develop?

- Why is it hard to develop software within budget and time?
- Quality of code
- When you say it's done, what do you mean?
- How good is the code you write?



### Efforts to minimize Risk

- Change in inevitable
- You don't want to wonder what the effect of a change is
- Feedback is critical
- Frequent feedback is necessary
- You want to know right away if you broke the code, isn't it?



#### Bridge Construction



- Safety Concerns
- Strong metrics and standards
- Often construction and design are separated
- Innovation and construction are separated







- In Software Development Construction is Cheap (it's the conversion of code into executables)
- Design (which involves modeling and coding) is expensive
- Can't we quickly test our design (since construction is cheap)?
- Testing is the Engineering Rigor in Software Development

SDC-10



## Waterfall-pros and cons

- Simple (simplistic)
- Easy to plan
- Hard to deliver
- Assumes stages carried out to completion
- Most practiced
- High rate of failure

SDC- 13

#### What's Agility?

- What's Agility?
  - It's being agile
- OK, what's Agile?
  - "marked by the ready ability to move with quick easy grace"
  - "having a quick resourceful and adaptive character"

SDC-14









Experienced project manager	15
Small milestones	10
	1

From Agile and Iterative Development: A Managers Guide by Craig Larman

SDC-20









# Communication Actively listen and seek feedback

- $\circ$
- Feedback comes in two forms
  - Is your code meeting and continuing to meet your (programmers') expectations?
    - Unit and integration tests
  - Is it relevant and solving customers' problems?  $\bigcirc$ 
    - Frequent Demo and Exercise

