

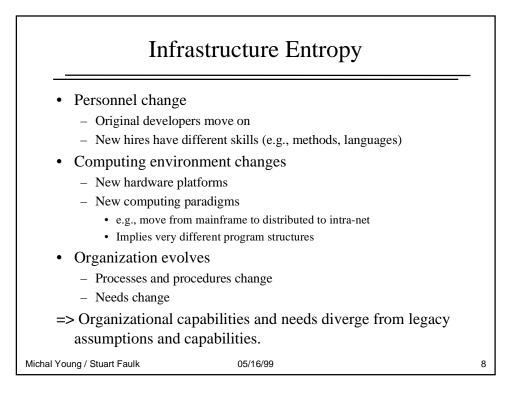
## How Software Rots

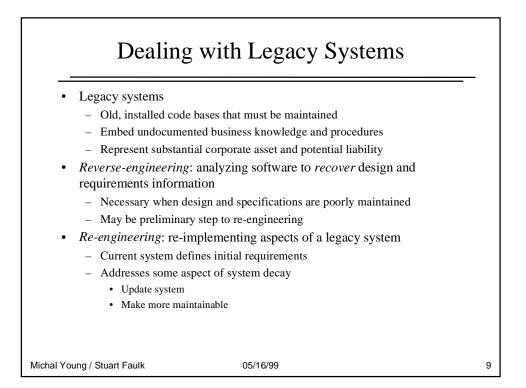
- Design is lost or out of date
- Comments are missing or wrong
- Each change makes it a little worse
  - Fossil code accumulates
  - "Secrets" leak out of modules
- Eventually there is no design, only an ecology of code
  - "What it should do" is replaced by "What it did before"
  - Bugs become features

Michal Young / Stuart Faulk

05/16/99

7





	an we make sense of a system It adequate documentation?	_
• ]	Reverse engineering / visualization	
	<ul> <li>Extract structural views from existing software, using static (and occasionally dynamic) analysis</li> </ul>	
	<ul> <li>Typically semi-automatic, analysis + user-controlled summarization. Main challenge is scale.</li> </ul>	
	- Examples: Rigi system, Murphy's reflexion models	
• (	Query systems	
	– Example: ISI natural language query system	

