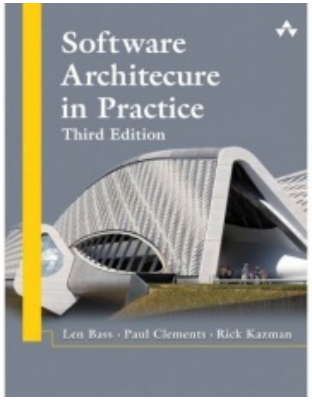
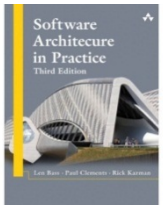
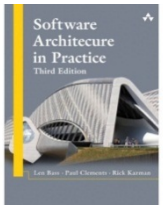


Chapter 11: Usability



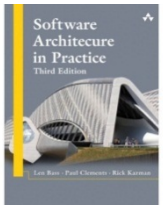
Chapter Outline

- What is Usability?
- Usability General Scenario
- Tactics for Usability
- A Design Checklist for Usability
- Summary



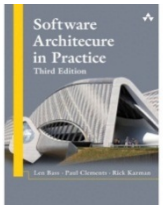
What is Usability?

- Usability is concerned with how easy it is for the user to accomplish a desired task and the kind of user support the system provides.
- Over the years, a focus on usability has shown itself to be one of the cheapest and easiest ways to improve a system's quality (or, more precisely, the user's *perception* of quality).



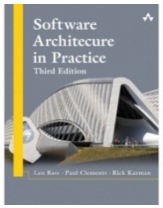
What is Usability?

- Usability comprises the following areas:
  - Learning system features.
  - Using a system efficiently.
  - Minimizing the impact of errors.
  - Adapting the system to user needs.
  - Increasing confidence and satisfaction.



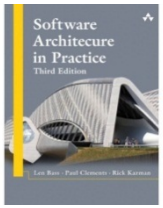
Usability General Scenario

Portion of Scenario	Possible Values
Source	End user, possibly in a specialized role
Stimulus	End user tries to use a system efficiently, learn to use the system, minimize the impact of errors, adapt the system, or configure the system
Environment	Runtime or configuration time
Artifacts	System or the specific portion of the system with which the user is interacting.
Response	The system should either provide the user with the features needed or anticipate the user's needs.
Response Measure	One or more of the following: task time, number of errors, number of tasks accomplished, user satisfaction, gain of user knowledge, ratio of successful operations to total operations, or amount of time or data lost when an error occurs.



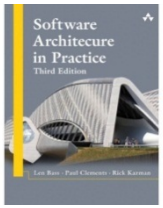
Sample Concrete Usability Scenario

- The user downloads a new application and is using it productively after two minutes of experimentation.

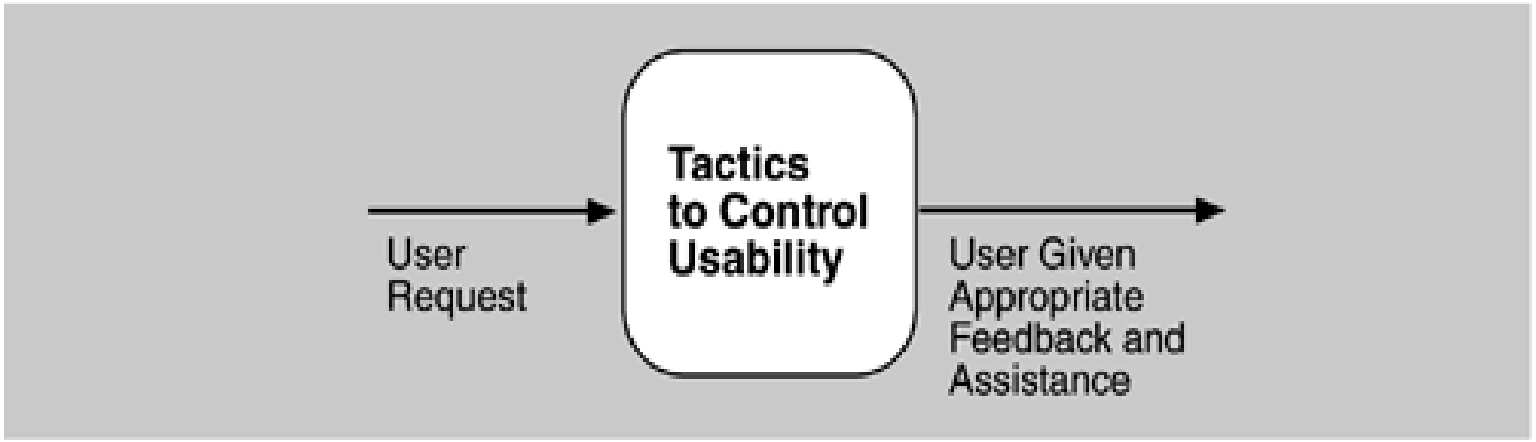


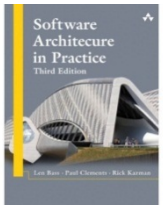
Goal of Usability Tactics

- Researchers in human-computer interaction have used the terms "user initiative," "system initiative," and "mixed initiative" to describe which of the human-computer pair takes the initiative in performing certain actions and how the interaction proceeds.
- Usability scenarios can combine initiatives from both perspectives.
- We use this distinction between user and system initiative to discuss the tactics that the architect uses to achieve the various scenarios.

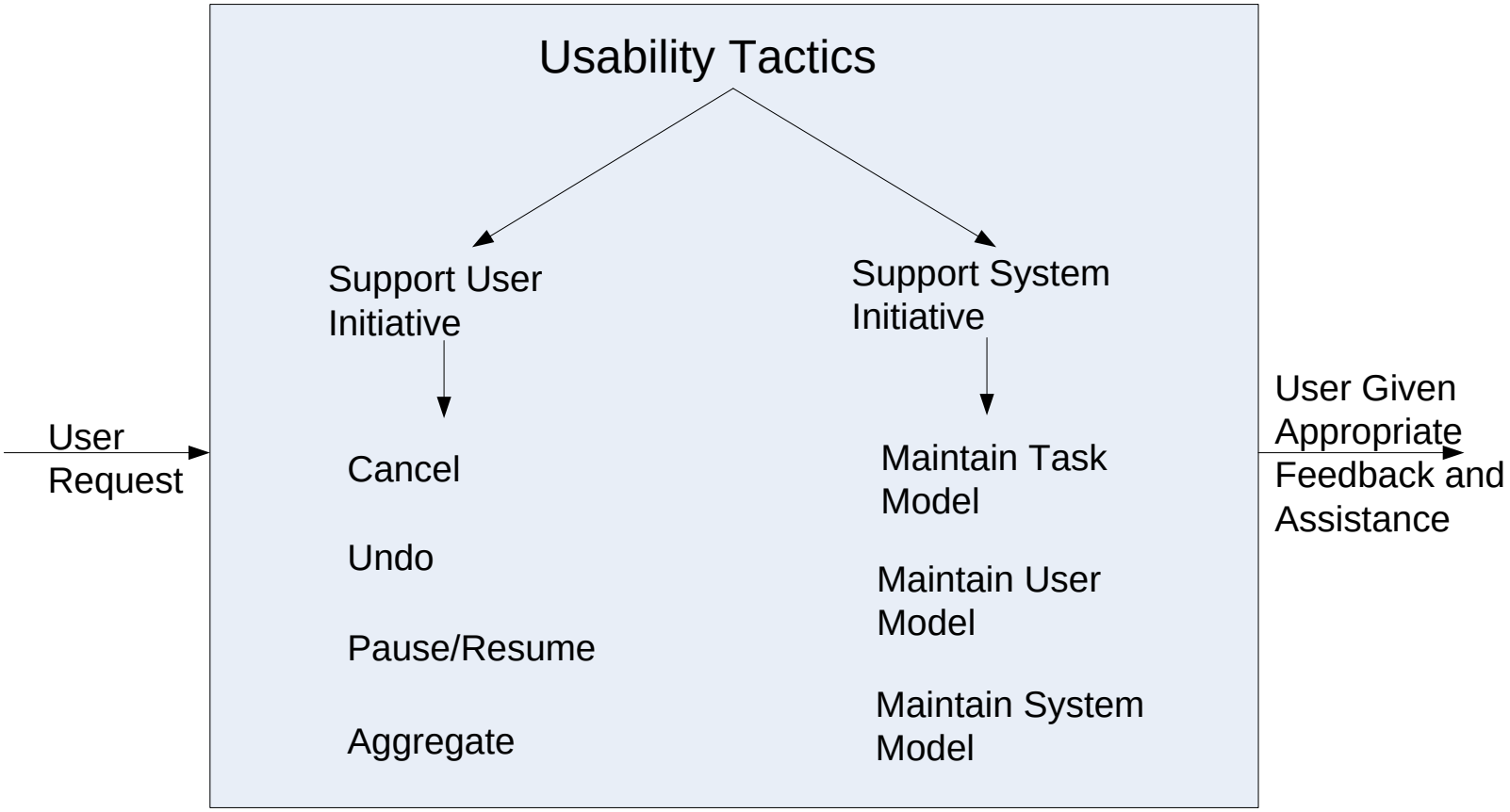


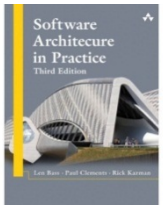
Goal of Usability Tactics





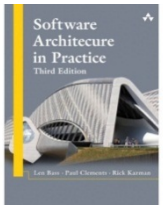
Usability Tactics





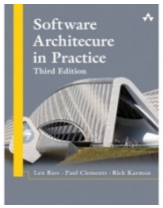
Support User Initiative

- Cancel: the system must listen for the cancel request; the command being canceled must be terminated; resources used must be freed; and collaborating components must be informed.
- Pause/Resume: temporarily free resources so that they may be re-allocated to other tasks.
- Undo: maintain a sufficient amount of information about system state so that an earlier state may be restored, at the user's request.
- Aggregate: ability to aggregate lower-level objects into a group, so that a user operation may be applied to the group, freeing the user from the drudgery.



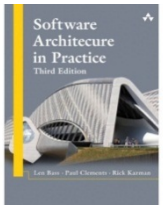
Support System Initiative

- Maintain Task Model: determines context so the system can have some idea of what the user is attempting and provide assistance.
- Maintain User Model: explicitly represents the user's knowledge of the system, the user's behavior in terms of expected response time, etc.
- Maintain System Model: system maintains an explicit model of itself. This is used to determine expected system behavior so that appropriate feedback can be given to the user.



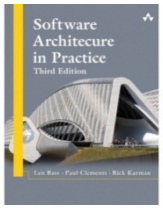
Design Checklist for Usability

Allocation of Responsibilities	**Ensure that additional system responsibilities have been allocated, as needed, to assist the user in** • **learning how to use the system** • **efficiently achieving the task at hand** • **adapting and configuring the system** • **recovering from user and system errors**



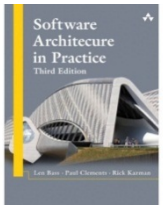
Design Checklist for Usability

Coordination Model	**Determine whether the properties of system elements' coordination—timeliness, currency, completeness, correctness, consistency—affect how a user learns to use the system, achieves goals or completes tasks, adapts and configures the system, recovers from user and system errors, increases confidence and satisfaction.** **For example, can the system respond to mouse events and give semantic feedback in real time? Can long-running events be canceled in a reasonable amount of time?**



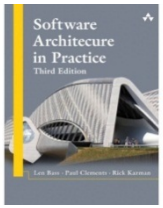
Design Checklist for Usability

Data Model	**Determine the major data abstractions that are involved with user-perceivable behavior.** **Ensure these major data abstractions, their operations, and their properties have been designed to assist the user in achieving the task at hand, adapting and configuring the system, recovering from user and system errors, learning how to use the system, and increasing satisfaction and user confidence** **For example, the data abstractions should be designed to support undo and cancel operations: the transaction granularity should not be so great that canceling or undoing an operation takes an excessively long time.**



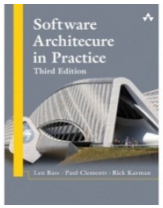
Design Checklist for Usability

Mapping Among Architectural Elements	**Determine what mapping among architectural elements is visible to the end user (for example, the extent to which the end user is aware of which services are local and which are remote).** **For those that are visible, determine how this affects the ways in which, or the ease with which the user will learn how to use the system, achieve the task at hand, adapt and configure the system, recover from user and system errors, and increase confidence and satisfaction.**



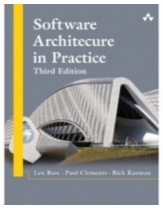
Design Checklist for Usability

Resource Management	**Determine how the user can adapt and configure the system's use of resources.** **Ensure that resource limitations under all user-controlled configurations will not make users less likely to achieve their tasks. For example, attempt to avoid configurations that would result in excessively long response times.** **Ensure that the level of resources will not affect the users' ability to learn how to use the system, or decrease their level of confidence and satisfaction with the system.**



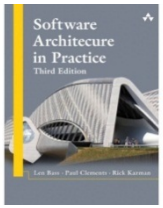
Design Checklist for Usability

Binding Time	**Determine which binding time decisions should be under user control and ensure that users can make decisions that aid in usability.** **For example, if the user can choose, at run-time, the system's configuration, or its communication protocols, or its functionality via plug-ins, you need to ensure that such choices do not adversely affect the user's ability to learn system features, use the system efficiently, minimize the impact of errors, further adapt and configure the system, or increase confidence and satisfaction.**



Design Checklist for Usability

Choice of Technology	**Ensure the chosen technologies help to achieve the usability scenarios that apply to your system. For example, do these technologies aid in the creation of on-line help, training materials, and user feedback collection.** **How usable are any of your chosen technologies? Ensure the chosen technologies do not adversely affect the usability of the system (in terms of learning system features, using the system efficiently, minimizing the impact of errors, or adapting/configuring the system, increase confidence and satisfaction).**



Summary

- Architectural support for usability involves both allowing the user to take the initiative in circumstances such as cancelling a long running command, undoing a completed command, and aggregating data and commands.
- To predict user or system response, the system must keep a model of the user, the system, and the task.