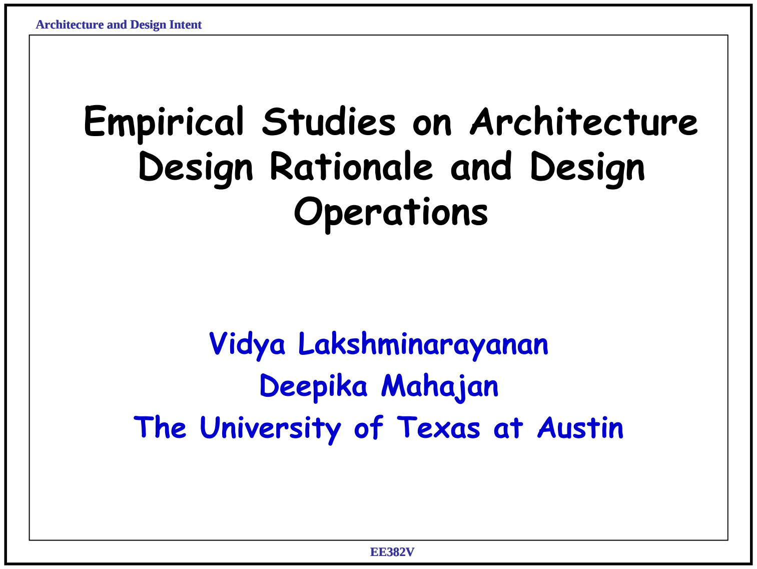
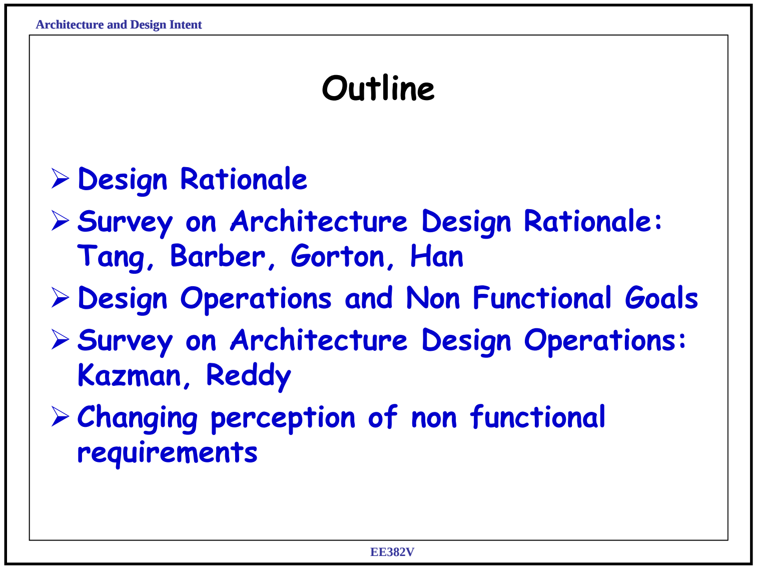


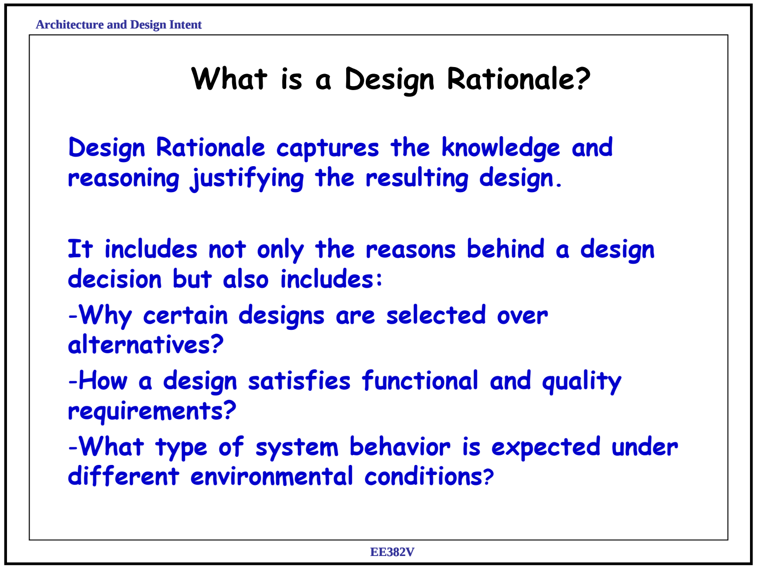
Empirical Studies on Architecture Design Rationale and Design Operations

Vidya Lakshminarayanan
Deepika Mahajan
The University of Texas at Austin



Outline

- Design Rationale
- Survey on Architecture Design Rationale: Tang, Barber, Gorton, Han
- Design Operations and Non Functional Goals
- Survey on Architecture Design Operations: Kazman, Reddy
- Changing perception of non functional requirements



What is a Design Rationale?

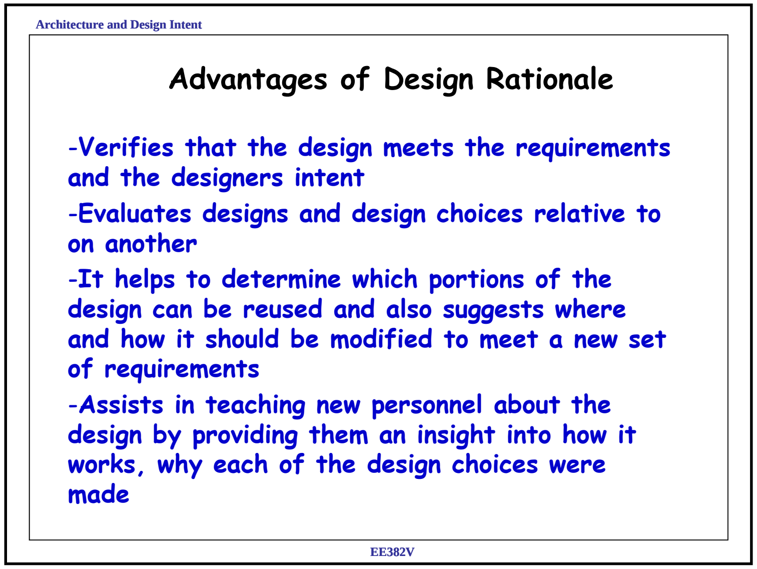
Design Rationale captures the knowledge and reasoning justifying the resulting design.

It includes not only the reasons behind a design decision but also includes:

-Why certain designs are selected over alternatives?

-How a design satisfies functional and quality requirements?

-What type of system behavior is expected under different environmental conditions?



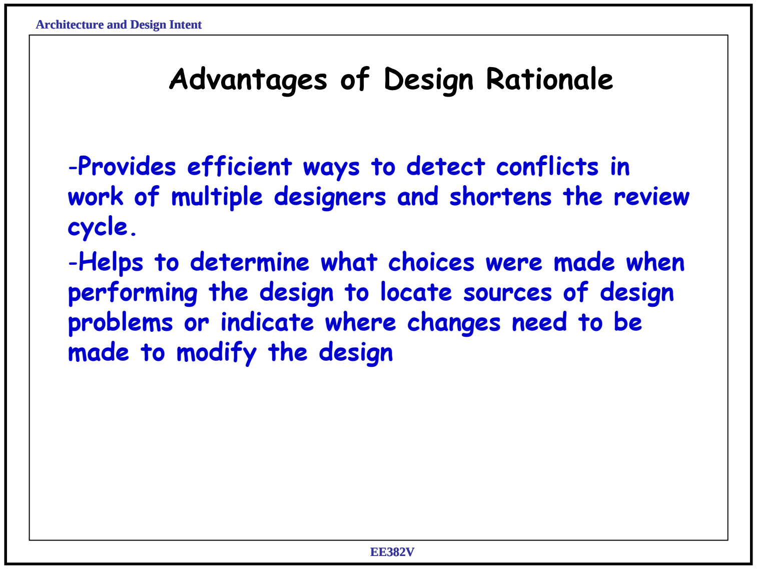
Advantages of Design Rationale

-Verifies that the design meets the requirements and the designers intent

-Evaluates designs and design choices relative to on another

-It helps to determine which portions of the design can be reused and also suggests where and how it should be modified to meet a new set of requirements

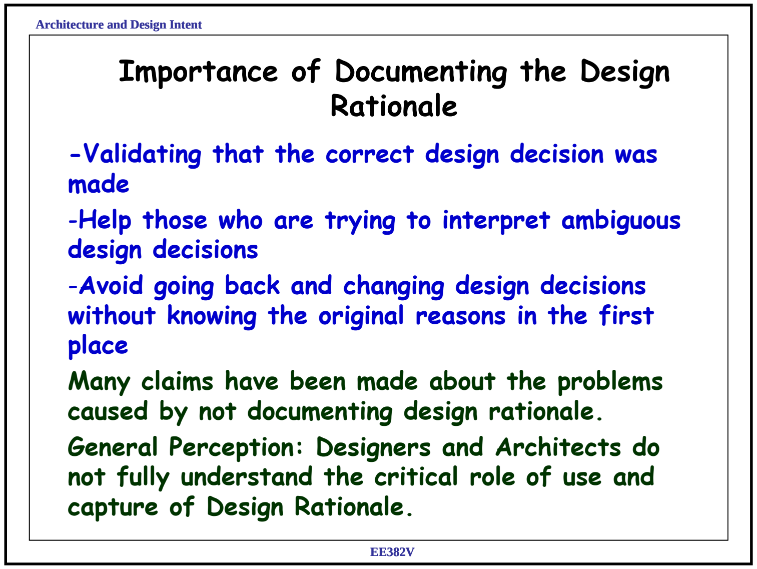
-Assists in teaching new personnel about the design by providing them an insight into how it works, why each of the design choices were made



Advantages of Design Rationale

-Provides efficient ways to detect conflicts in work of multiple designers and shortens the review cycle.

-Helps to determine what choices were made when performing the design to locate sources of design problems or indicate where changes need to be made to modify the design



Importance of Documenting the Design Rationale

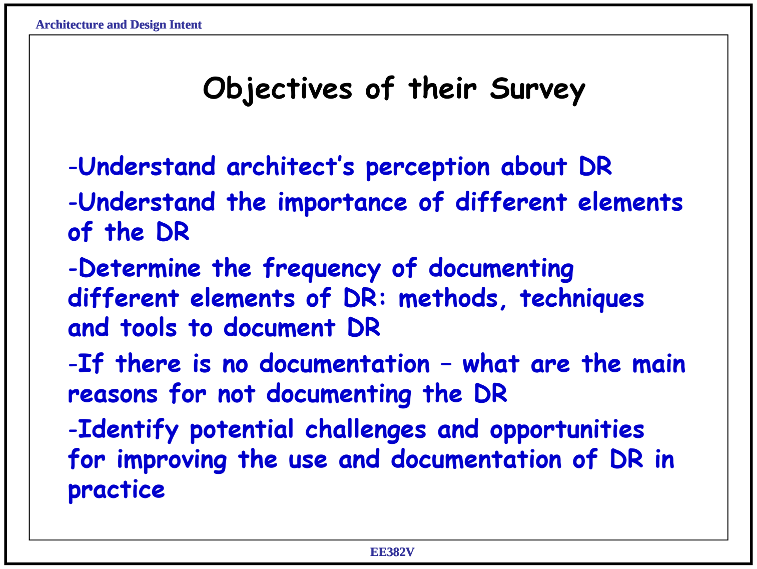
-Validating that the correct design decision was made

-Help those who are trying to interpret ambiguous design decisions

-Avoid going back and changing design decisions without knowing the original reasons in the first place

Many claims have been made about the problems caused by not documenting design rationale.

General Perception: Designers and Architects do not fully understand the critical role of use and capture of Design Rationale.



Objectives of their Survey

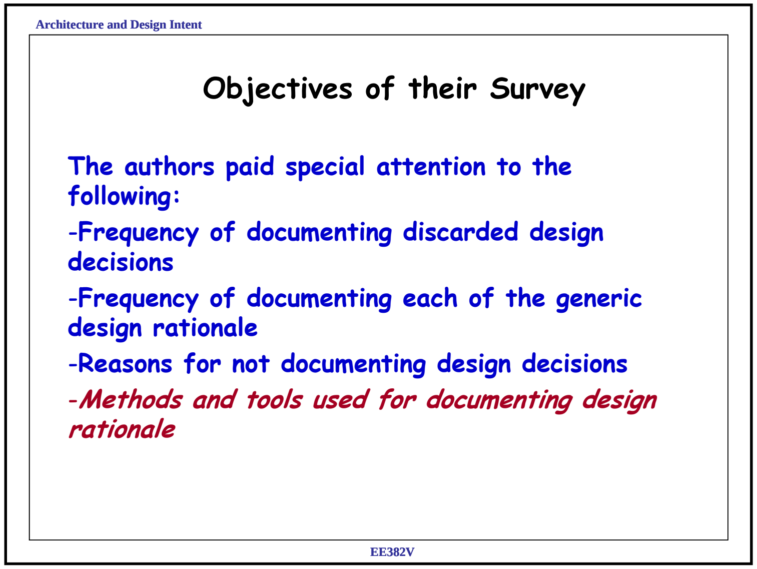
-Understand architect's perception about DR

-Understand the importance of different elements of the DR

-Determine the frequency of documenting different elements of DR: methods, techniques and tools to document DR

-If there is no documentation – what are the main reasons for not documenting the DR

-Identify potential challenges and opportunities for improving the use and documentation of DR in practice



Objectives of their Survey

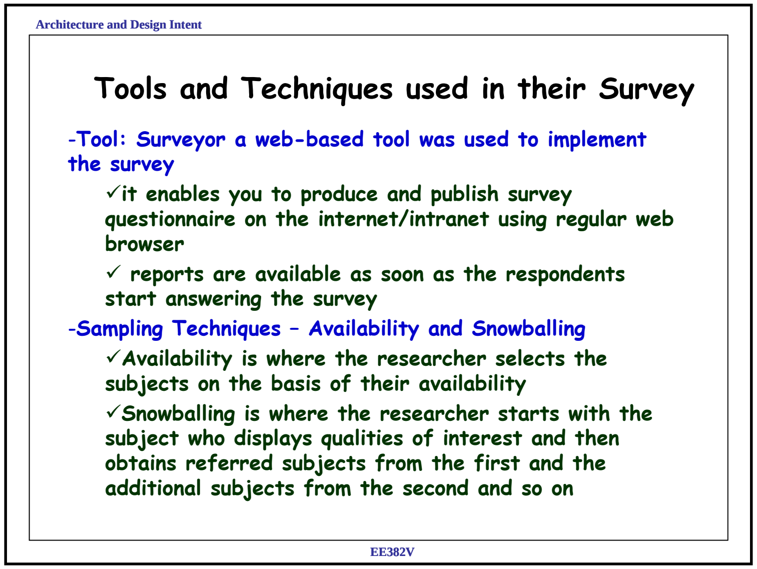
The authors paid special attention to the following:

-Frequency of documenting discarded design decisions

-Frequency of documenting each of the generic design rationale

-Reasons for not documenting design decisions

-Methods and tools used for documenting design rationale



Tools and Techniques used in their Survey

-Tool: Surveyor a web-based tool was used to implement the survey

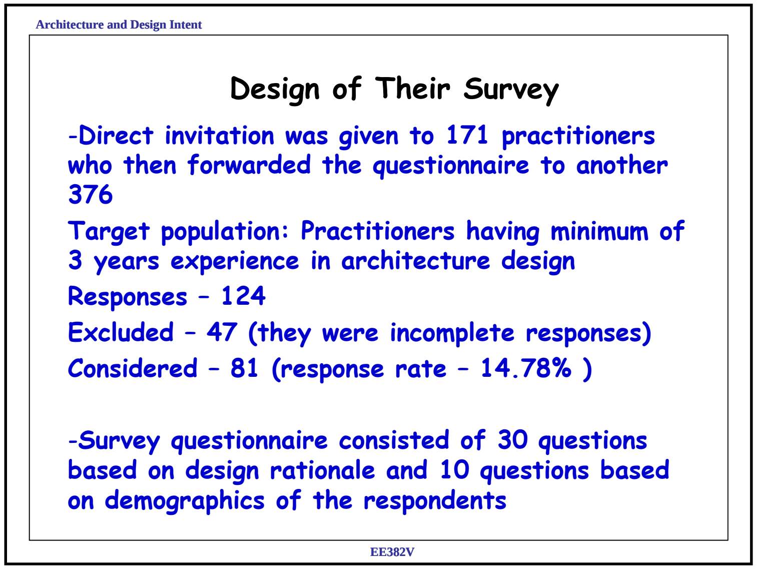
✓it enables you to produce and publish survey questionnaire on the internet/intranet using regular web browser

✓ reports are available as soon as the respondents start answering the survey

-Sampling Techniques – Availability and Snowballing

✓Availability is where the researcher selects the subjects on the basis of their availability

✓Snowballing is where the researcher starts with the subject who displays qualities of interest and then obtains referred subjects from the first and the additional subjects from the second and so on



Design of Their Survey

-Direct invitation was given to 171 practitioners who then forwarded the questionnaire to another 376

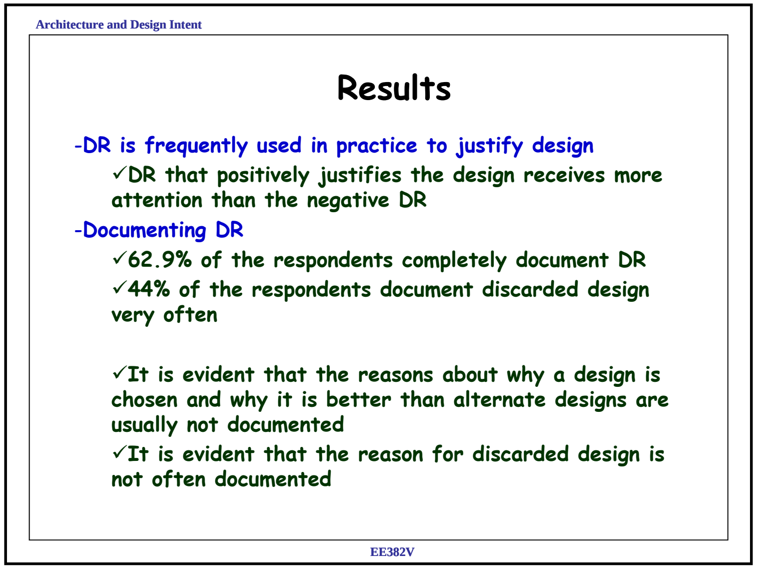
Target population: Practitioners having minimum of 3 years experience in architecture design

Responses – 124

Excluded – 47 (they were incomplete responses)

Considered – 81 (response rate – 14.78%)

-Survey questionnaire consisted of 30 questions based on design rationale and 10 questions based on demographics of the respondents



Results

-DR is frequently used in practice to justify design

✓DR that positively justifies the design receives more attention than the negative DR

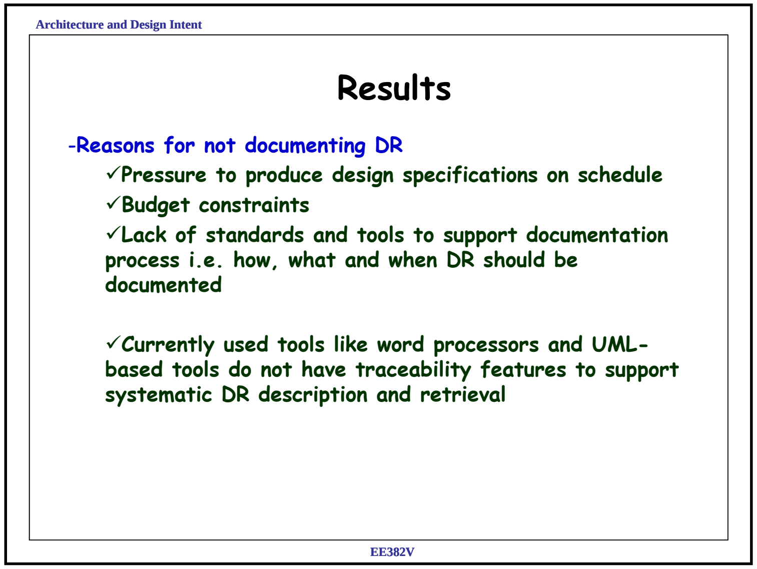
-Documenting DR

✓62.9% of the respondents completely document DR

✓44% of the respondents document discarded design very often

✓It is evident that the reasons about why a design is chosen and why it is better than alternate designs are usually not documented

✓It is evident that the reason for discarded design is not often documented



Results

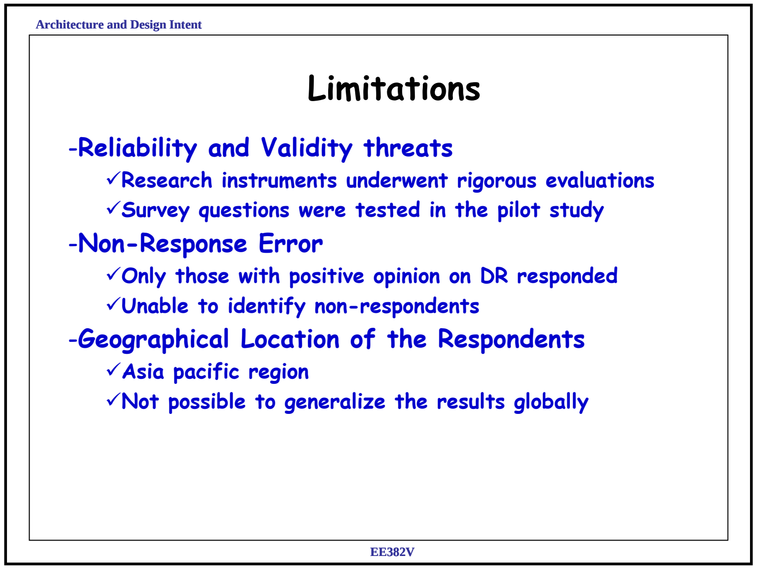
-Reasons for not documenting DR

✓Pressure to produce design specifications on schedule

✓Budget constraints

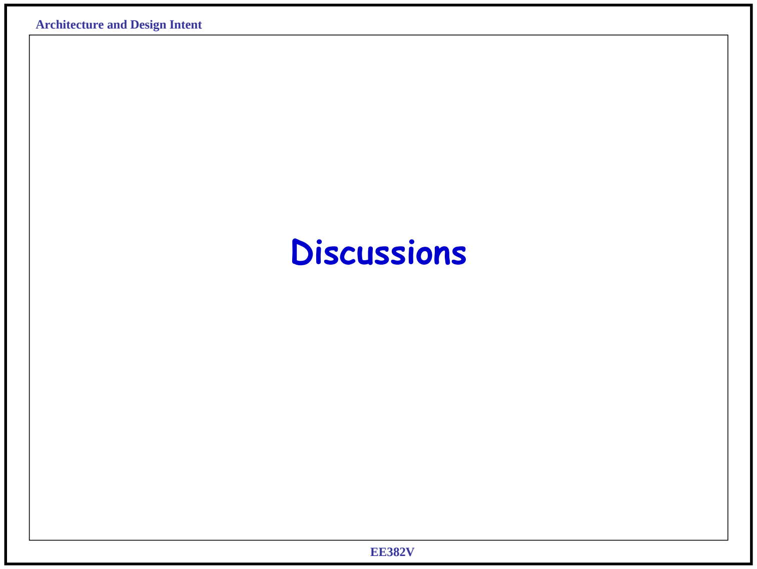
✓Lack of standards and tools to support documentation process i.e. how, what and when DR should be documented

✓Currently used tools like word processors and UML-based tools do not have traceability features to support systematic DR description and retrieval

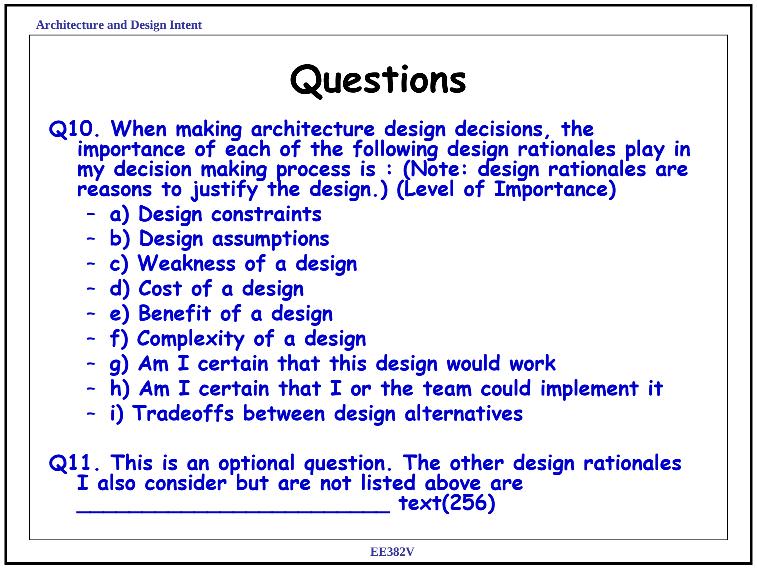


Limitations

- -Reliability and Validity threats
  - ✓Research instruments underwent rigorous evaluations
  - ✓Survey questions were tested in the pilot study
- -Non-Response Error
  - ✓Only those with positive opinion on DR responded
  - ✓Unable to identify non-respondents
- -Geographical Location of the Respondents
  - ✓Asia pacific region
  - ✓Not possible to generalize the results globally



Discussions

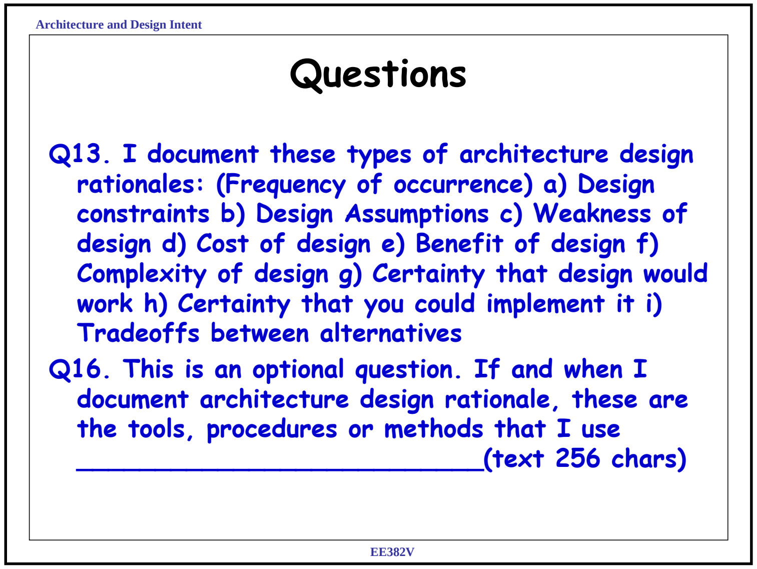


Questions

Q10. When making architecture design decisions, the importance of each of the following design rationales play in my decision making process is : (Note: design rationales are reasons to justify the design.) (Level of Importance)

- a) Design constraints
- b) Design assumptions
- c) Weakness of a design
- d) Cost of a design
- e) Benefit of a design
- f) Complexity of a design
- g) Am I certain that this design would work
- h) Am I certain that I or the team could implement it
- i) Tradeoffs between design alternatives

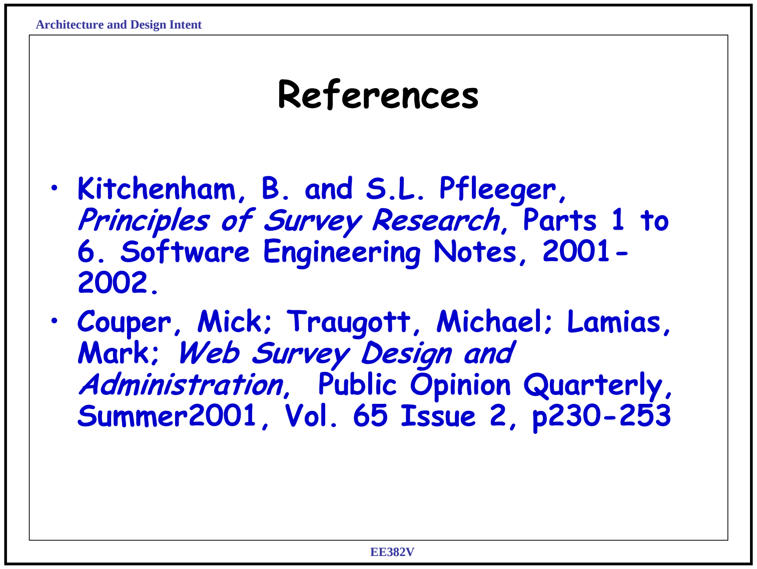
Q11. This is an optional question. The other design rationales I also consider but are not listed above are ______________________________ text(256)



Questions

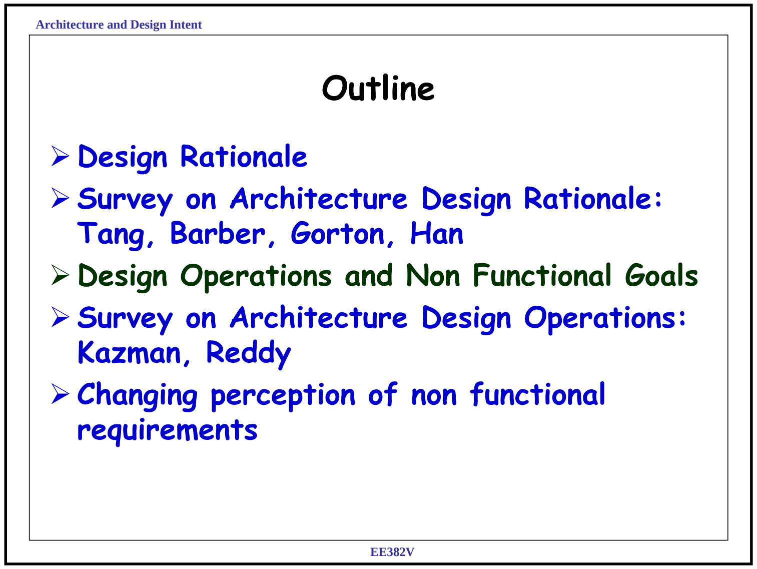
Q13. I document these types of architecture design rationales: (Frequency of occurrence) a) Design constraints b) Design Assumptions c) Weakness of design d) Cost of design e) Benefit of design f) Complexity of design g) Certainty that design would work h) Certainty that you could implement it i) Tradeoffs between alternatives

Q16. This is an optional question. If and when I document architecture design rationale, these are the tools, procedures or methods that I use ______________________________(text 256 chars)



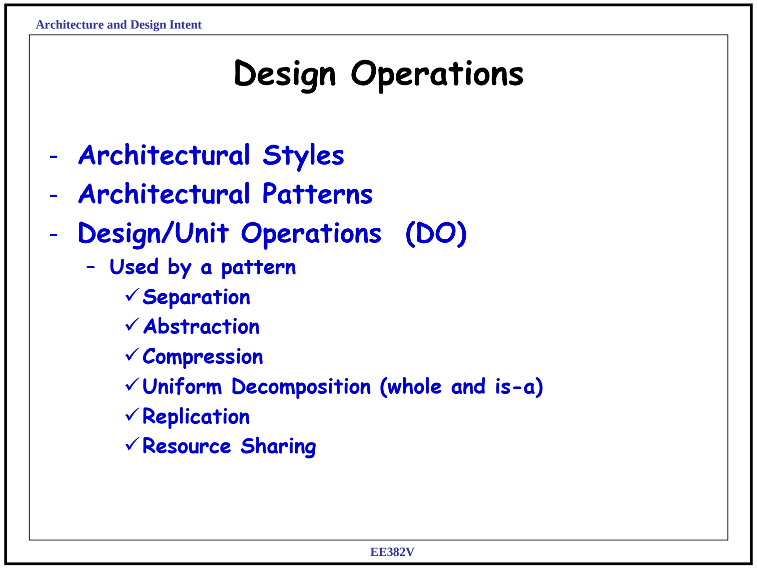
References

- Kitchenham, B. and S.L. Pfleeger, *Principles of Survey Research*, Parts 1 to 6. Software Engineering Notes, 2001-2002.
- Couper, Mick; Traugott, Michael; Lamias, Mark; *Web Survey Design and Administration*, Public Opinion Quarterly, Summer2001, Vol. 65 Issue 2, p230-253



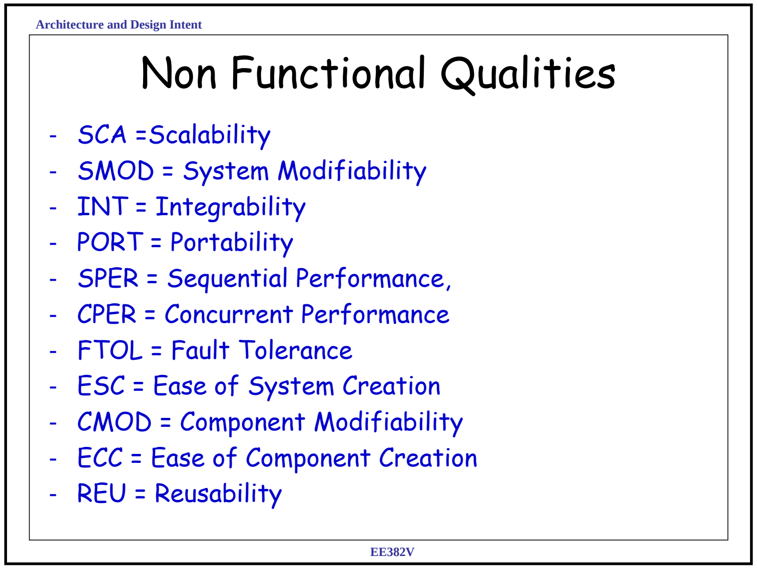
Outline

- ➢ Design Rationale
- ➢ Survey on Architecture Design Rationale: Tang, Barber, Gorton, Han
- ➢ Design Operations and Non Functional Goals
- ➢ Survey on Architecture Design Operations: Kazman, Reddy
- ➢ Changing perception of non functional requirements



Design Operations

- Architectural Styles
- Architectural Patterns
- Design/Unit Operations (DO)
  - Used by a pattern
    - ✓Separation
    - ✓Abstraction
    - ✓Compression
    - ✓Uniform Decomposition (whole and is-a)
    - ✓Replication
    - ✓Resource Sharing



Non Functional Qualities

- SCA =Scalability
- SMOD = System Modifiability
- INT = Integrability
- PORT = Portability
- SPER = Sequential Performance,
- CPER = Concurrent Performance
- FTOL = Fault Tolerance
- ESC = Ease of System Creation
- CMOD = Component Modifiability
- ECC = Ease of Component Creation
- REU = Reusability

Unit Operations & Non Functional Qualities

- **Separation**
  - ✓ Distinct functionality ➔ Distinct component
  - ✓ Superset of the other unit operations
  - ✓ Permits parallelism, aids modifiability and portability
- **Abstraction**
  - ✓ Creation of a VM/VI: hide the underlying implementation
    - Emulate non native functionality
    - Layered systems: ISO/OSI
  - ✓ Hard to create, but can be re-used by other software systems ➔ subsequent ease of creation and maintenance
- **Compression**
  - ✓ Removing of layers or interfaces that separate system functions
  - ✓ Improve system performance, to speed system development

Unit Operations & Non Functional Qualities

- **Uniform Decomposition**
  - ✓ Part-whole and Is-a
  - ✓ Eases integration and scaling of the system
- **Replication**
  - ✓ Duplicate component within an architecture
  - ✓ Enhances reliability and performance
    - Redundancy ➔ Reliability
    - Performance ➔ Increased parallelism
    - Tradeoff: Increased throughput, lower MTTF
- **Resource Sharing**
  - ✓ Enhances integrability, portability, modifiability

Non Functional Requirements and Design Operations

- Non functional requirements
  - ✓ Not addressed in design
    - Recent research on current industry practices suggests otherwise
  - ✓ Addressed through non architectural means such as coding or testing
  - ✓ Larger systems: non functional requirements are more complex, sometimes conflicting, harder to identify trade-offs
- Consider non-functional requirements simultaneously
  - ✓ Design Operations (DO)

Outline

- ➢ Design Rationale
- ➢ Survey on Architecture Design Rationale: Tang, Barber, Gorton, Han
- ➢ Design Operations and Non Functional Goals
- ➢ Survey on Architecture Design Operations: Kazman, Reddy
  - ➢ Objectives & Design
  - ➢ Results & Limitations
- ➢ Changing perception of non functional requirements

Survey: Objectives

- Empirically validate a set of Design Operations which are used to achieve non functional quality goals
  - ✓ "Statistical measures of a design operation's impact on promoting or inhibiting the achievement of a non-functional requirement, within the context of a particular development scenario"

Survey: Design

- Designers asked to identify the relationship between a DO and a number of non functional qualities
  - ✓ Scalability, integrability, portability, sequential performance, concurrent performance, fault tolerance, system creation & modifiability, component creation & modifiability, reusability
- 21 experts from 9 companies, 66 questions
- All questions were structured around a miniature system development example

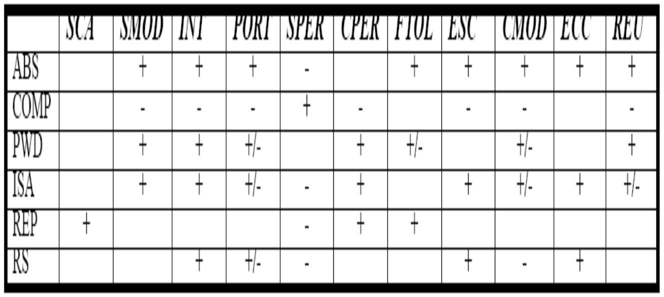
Survey: Results

- Statistical measure of degree of agreement among respondents where the answers fall on an ordinal scale:

$$V = s/m^2 \times (s-1)\sum_j (x_i - m/s)^2$$

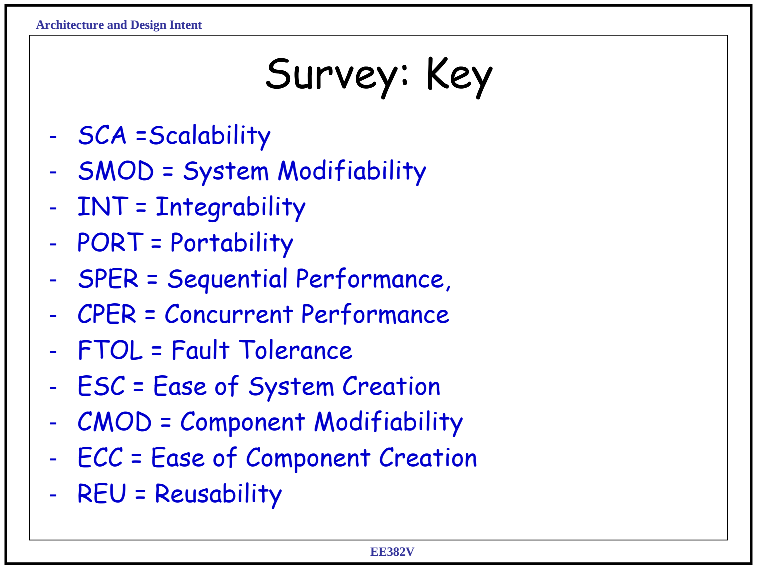
✓ *xj* = the number of respondents who assigned the *j*-th rank to the question
✓ *s* = the number of ranks on the scale
✓ *m* = the number of respondents for the question
✓ *V* = a range from 0 to 1

Survey: Results



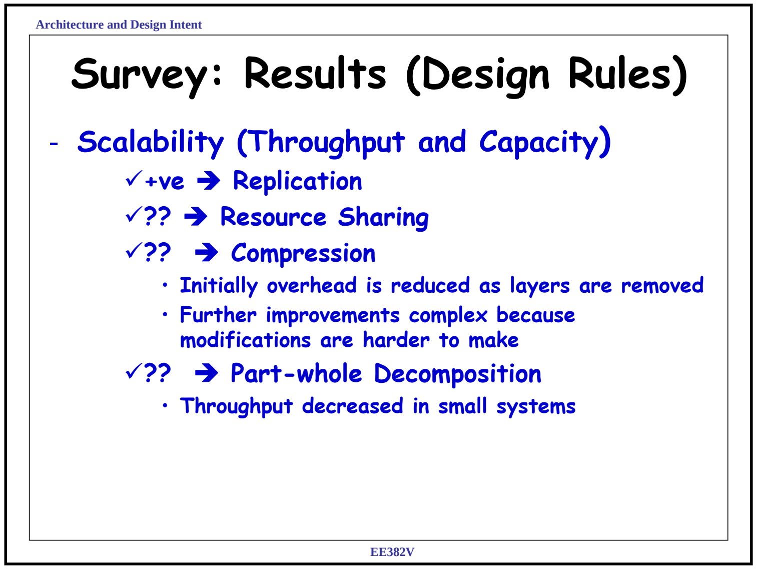
	SCA	SMOD	INT	PORT	SPER	CPER	FTOL	ESC	CMOD	ECC	REU
ABS		+	+	+	-		+	+	+	+	+
COMP		-	-	-	+	-		-	-		-
PWD		+	+	+/-		+	+/-		+/-		+
ISA		+	+	+/-	-	+		+	+/-	+	+/-
REP	+				-	+	+				
RS			+	+/-	-			+	-	+	

Table 2: Summary of Relationships between Unit Operations and Qualities



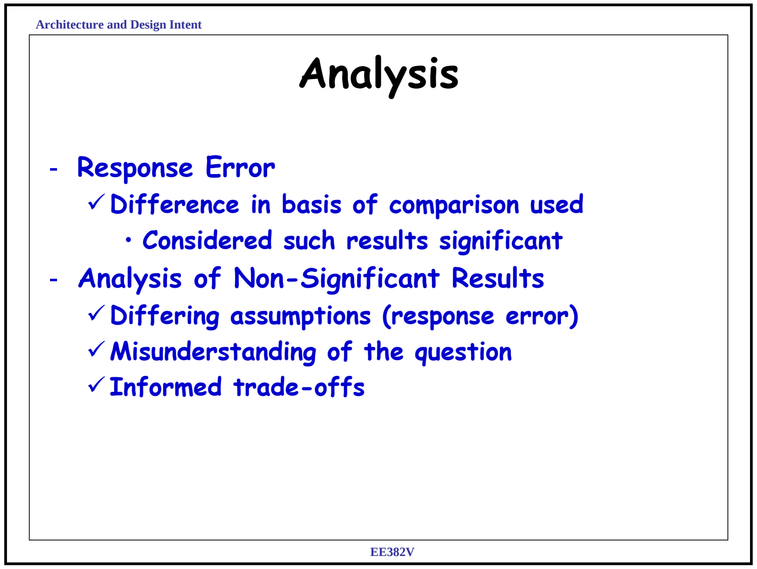
Survey: Key

- SCA =Scalability
- SMOD = System Modifiability
- INT = Integrability
- PORT = Portability
- SPER = Sequential Performance,
- CPER = Concurrent Performance
- FTOL = Fault Tolerance
- ESC = Ease of System Creation
- CMOD = Component Modifiability
- ECC = Ease of Component Creation
- REU = Reusability



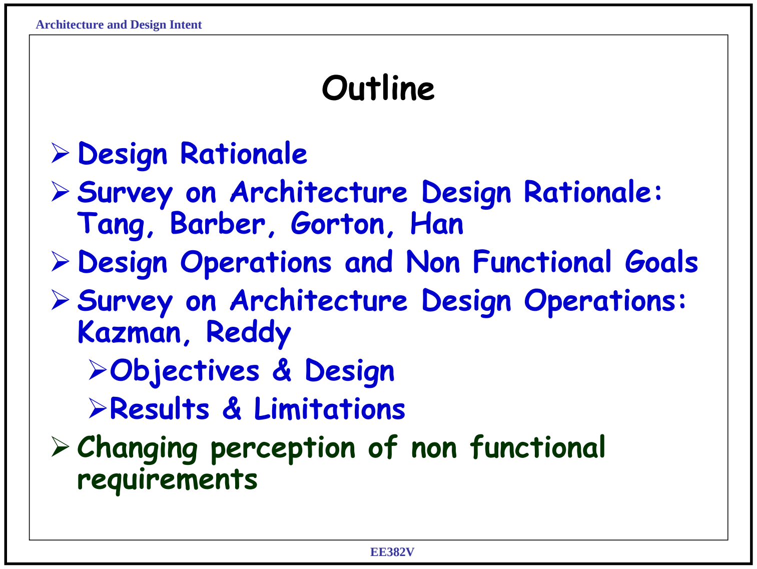
Survey: Results (Design Rules)

- Scalability (Throughput and Capacity)
  - ✓+ve ➔ Replication
  - ✓?? ➔ Resource Sharing
  - ✓?? ➔ Compression
    - Initially overhead is reduced as layers are removed
    - Further improvements complex because modifications are harder to make
  - ✓?? ➔ Part-whole Decomposition
    - Throughput decreased in small systems



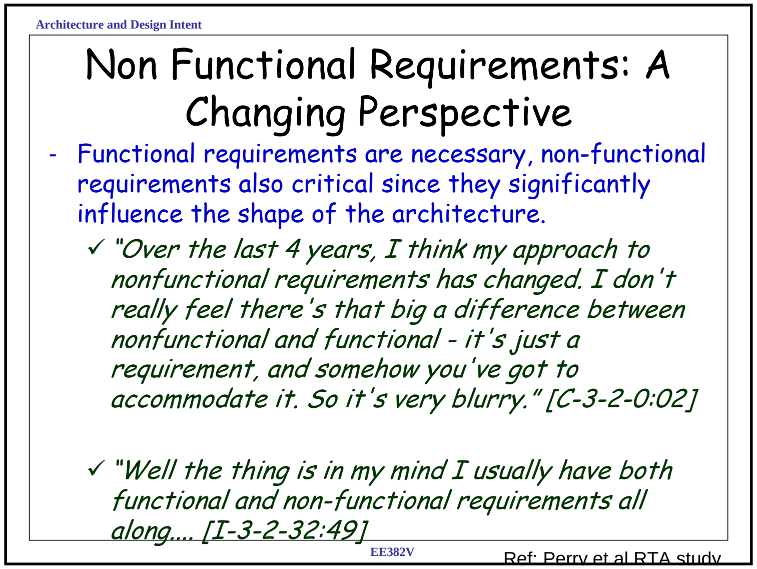
Analysis

- Response Error
  - ✓Difference in basis of comparison used
    - Considered such results significant
- Analysis of Non-Significant Results
  - ✓Differing assumptions (response error)
  - ✓Misunderstanding of the question
  - ✓Informed trade-offs



Outline

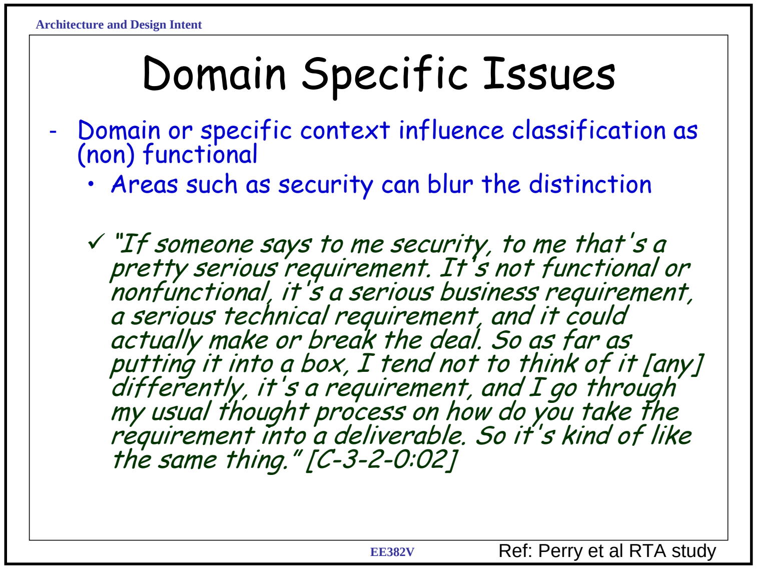
- ➢ Design Rationale
- ➢ Survey on Architecture Design Rationale: Tang, Barber, Gorton, Han
- ➢ Design Operations and Non Functional Goals
- ➢ Survey on Architecture Design Operations: Kazman, Reddy
  - ➢Objectives & Design
  - ➢Results & Limitations
- ➢ Changing perception of non functional requirements



Non Functional Requirements: A Changing Perspective

- Functional requirements are necessary, non-functional requirements also critical since they significantly influence the shape of the architecture.
  - ✓ *"Over the last 4 years, I think my approach to nonfunctional requirements has changed. I don't really feel there's that big a difference between nonfunctional and functional - it's just a requirement, and somehow you've got to accommodate it. So it's very blurry." [C-3-2-0:02]*
  - ✓ *"Well the thing is in my mind I usually have both functional and non-functional requirements all along.... [I-3-2-32:49]*

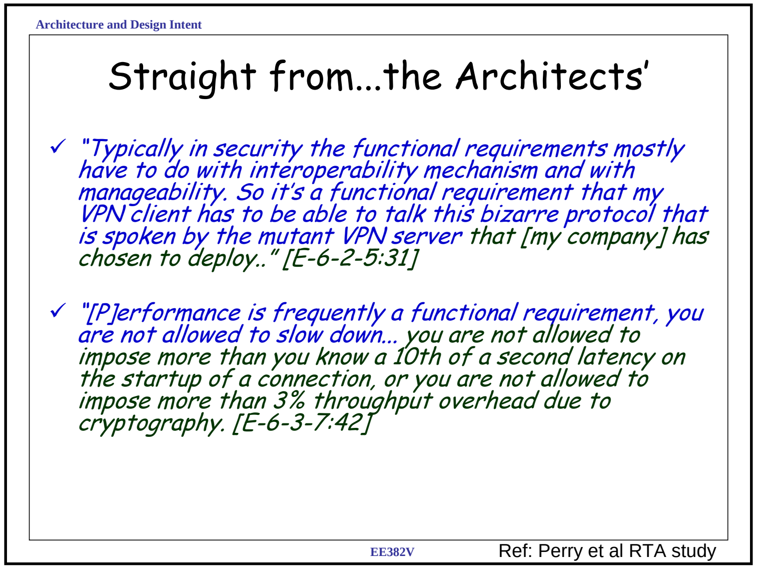
Ref: Perry et al RTA study



Domain Specific Issues

- Domain or specific context influence classification as (non) functional
  - Areas such as security can blur the distinction
  - ✓ *"If someone says to me security, to me that's a pretty serious requirement. It's not functional or nonfunctional, it's a serious business requirement, a serious technical requirement, and it could actually make or break the deal. So as far as putting it into a box, I tend not to think of it [any] differently, it's a requirement, and I go through my usual thought process on how do you take the requirement into a deliverable. So it's kind of like the same thing." [C-3-2-0:02]*

Ref: Perry et al RTA study

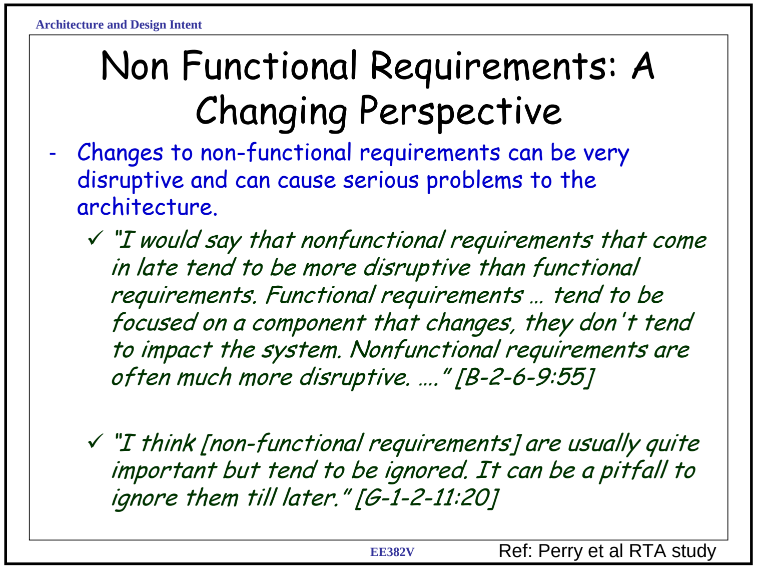


Straight from...the Architects'

✓ *"Typically in security the functional requirements mostly have to do with interoperability mechanism and with manageability. So it's a functional requirement that my VPN client has to be able to talk this bizarre protocol that is spoken by the mutant VPN server that [my company] has chosen to deploy.." [E-6-2-5:31]*

✓ *"[P]erformance is frequently a functional requirement, you are not allowed to slow down... you are not allowed to impose more than you know a 10th of a second latency on the startup of a connection, or you are not allowed to impose more than 3% throughput overhead due to cryptography. [E-6-3-7:42]*

Ref: Perry et al RTA study



Non Functional Requirements: A Changing Perspective

- Changes to non-functional requirements can be very disruptive and can cause serious problems to the architecture.
  - ✓ *"I would say that nonfunctional requirements that come in late tend to be more disruptive than functional requirements. Functional requirements … tend to be focused on a component that changes, they don't tend to impact the system. Nonfunctional requirements are often much more disruptive. …." [B-2-6-9:55]*
  - ✓ *"I think [non-functional requirements] are usually quite important but tend to be ignored. It can be a pitfall to ignore them till later." [G-1-2-11:20]*

Ref: Perry et al RTA study

Conclusion

- Design Operations
- Discussion of Survey
  - ✓Technical Report with details of the survey and results: MIA!
  - ✓"experience" of respondents
- Changing perception of non functional requirements

Questions?

Additional Material

Architecture of system with Abstraction

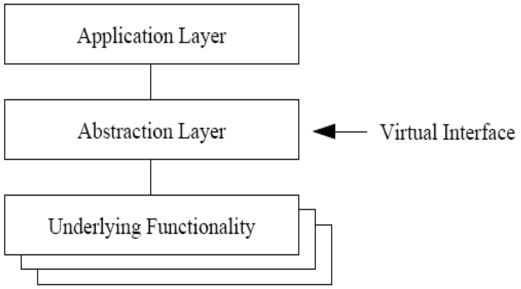
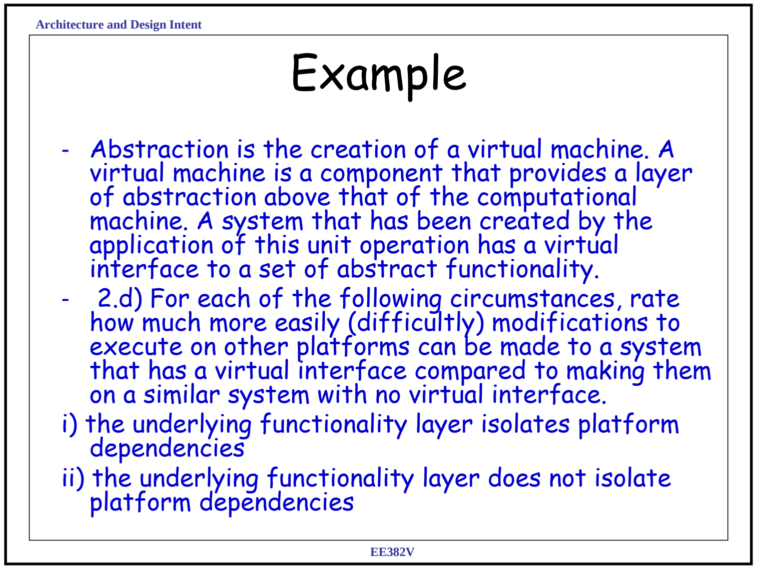


Figure 1: A System with a Virtual Interface



Architecture and Design Intent

Example

- Abstraction is the creation of a virtual machine. A virtual machine is a component that provides a layer of abstraction above that of the computational machine. A system that has been created by the application of this unit operation has a virtual interface to a set of abstract functionality.
- 2.d) For each of the following circumstances, rate how much more easily (difficultly) modifications to execute on other platforms can be made to a system that has a virtual interface compared to making them on a similar system with no virtual interface.

i) the underlying functionality layer isolates platform dependencies

ii) the underlying functionality layer does not isolate platform dependencies

EE382V