Using Quality Requirements to Systematically Develop Quality Software

Lawrence Chung
University of Texas, Dallas

Brian A. Nixon

Eric Yu

Dept. of Computer Science
University of Toronto

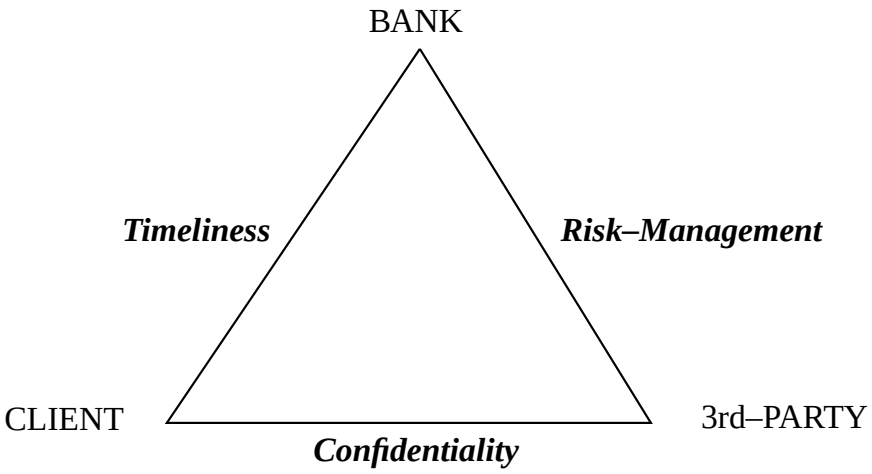
Fourth International Conference on Software Quality
McLean, VA, U.S.A. October 3–5, 1994.

In the Banking World ...

Can we build quality into software?

Update & Display Accounts
(Functional Requirements)
in an Accurate, Secure, fast, user-friendly manner
(Quality Requirements — Non-Functional Requirements)

In the Banking World ...



Can we build quality into software?

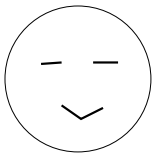
Software Development: *Art* → *Engineering*

Software Crafting →

Requirements
- Ambiguous
- Incomplete
- Inconsistent

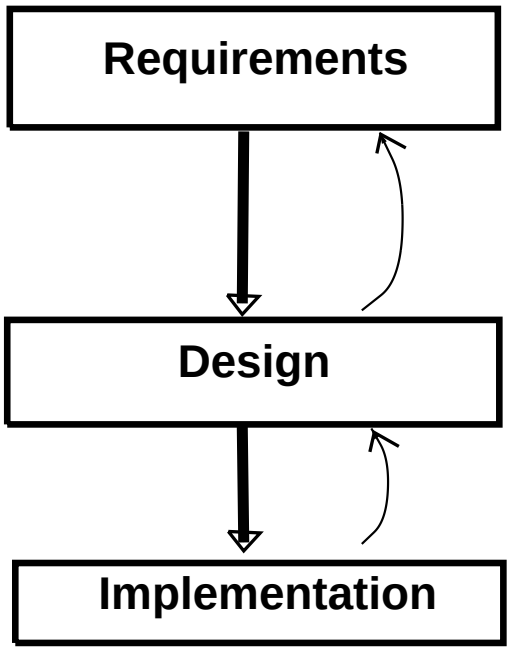
Product
- Inconsistent
- Untraceable
- Unjustifiable
- Unevolvable

Software Engineering →



disciplines, models, methodologies, tools

Software Life-Cycle Includes



Why **Quality Requirements?**

Quality is fitness for use — Juran

freedom from deficiency

Quality is free — Crosby

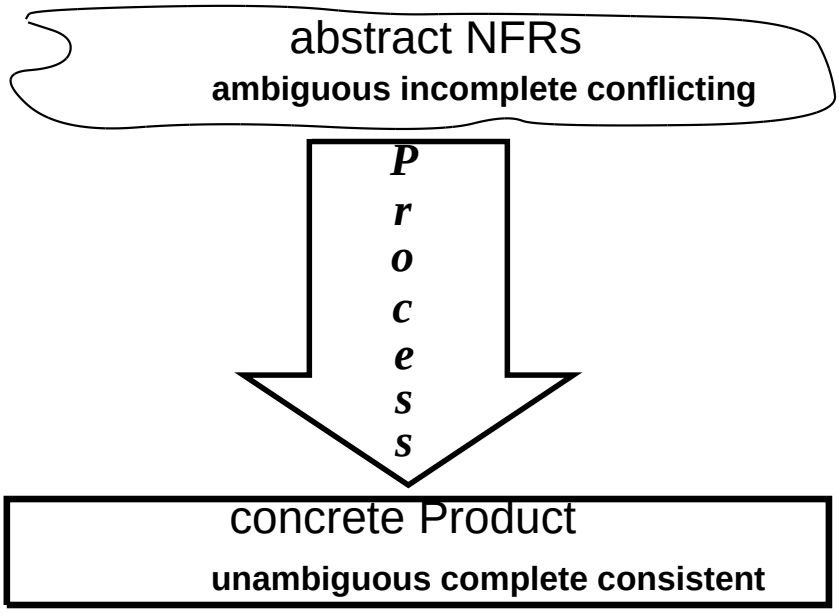
get the "right" requirements &

do it "right" the first time

Defects are costly — Boehm

\$ product errors = 100 × \$ requirements errors

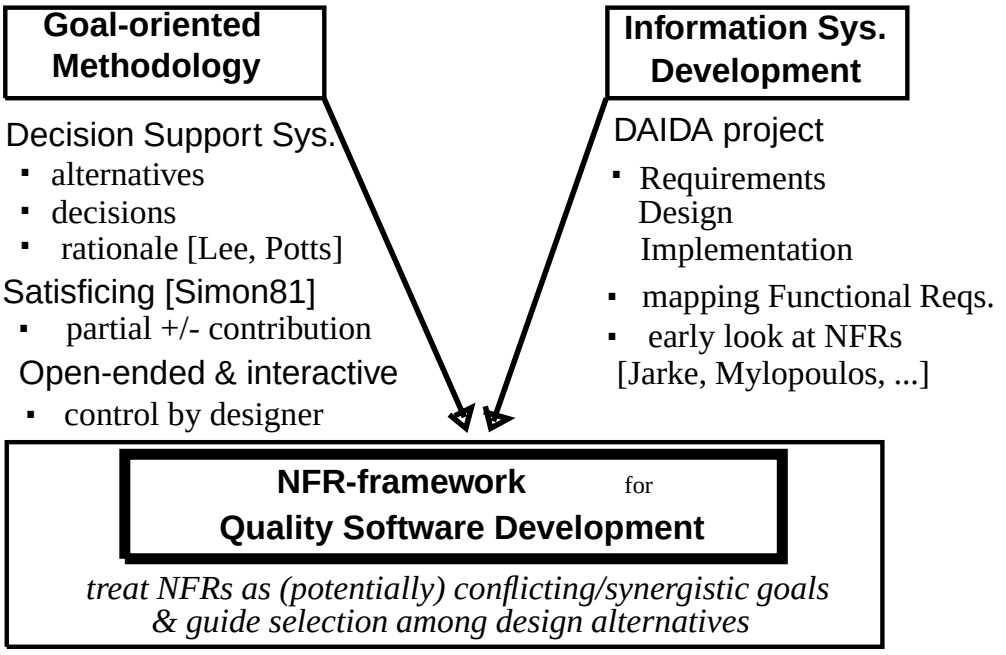
Quality Product — *How?*



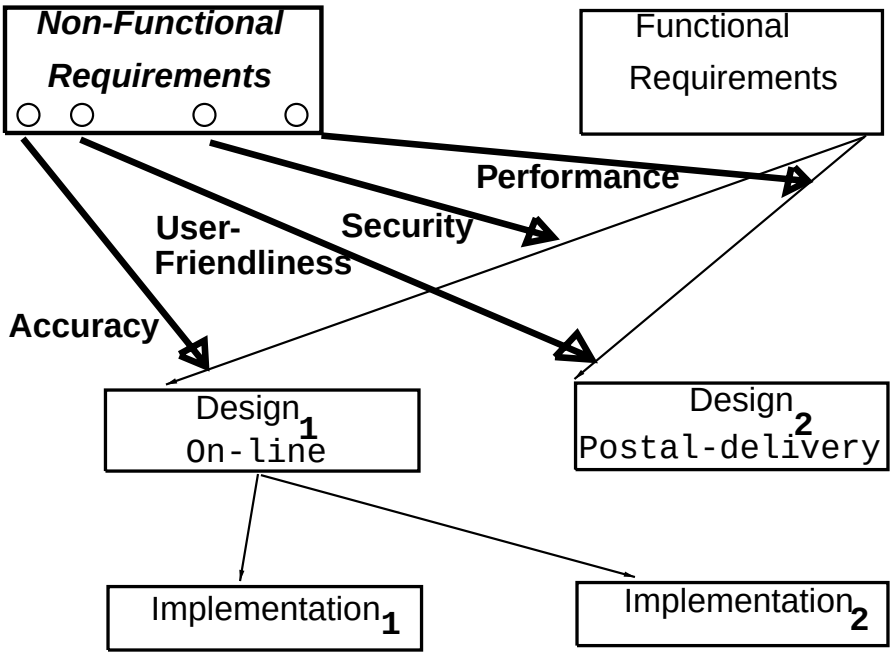
Outline

- Motivation
- **Framework**
- **Development Tool**
- **Three Small Applications**
- Conclusions

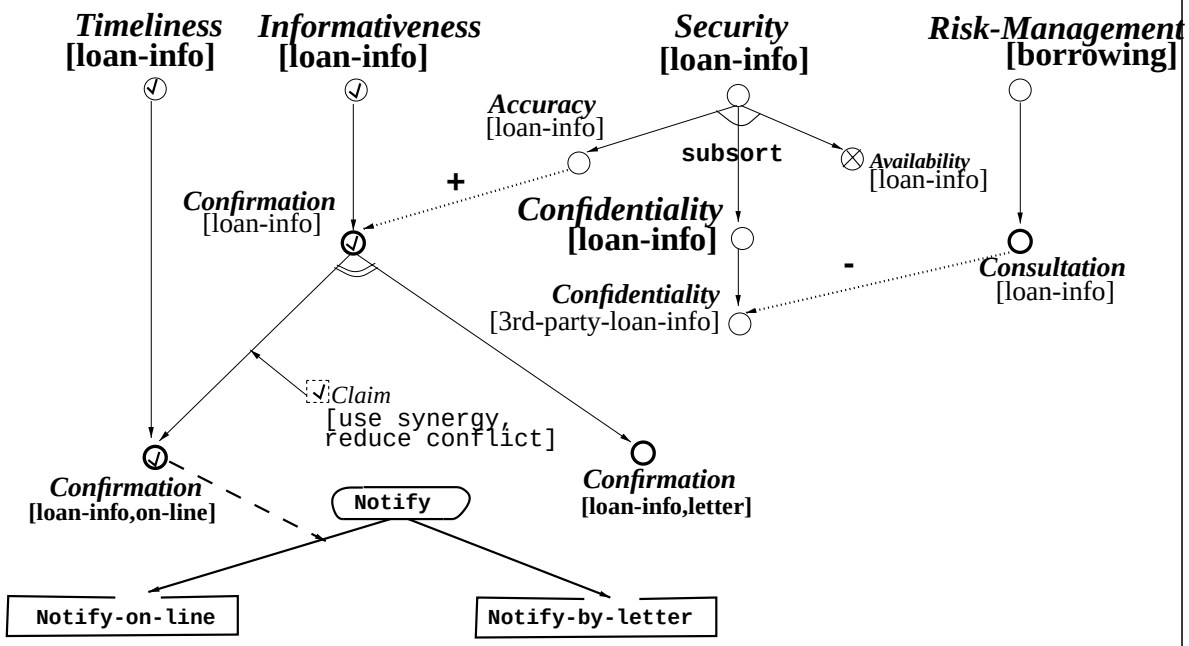
NFR framework: Approach



Goal-driven, Qualitative, Argumentative Process

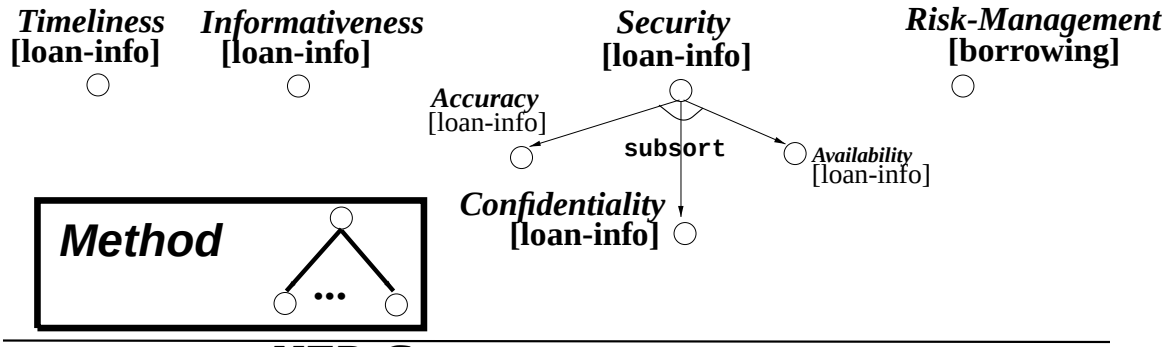


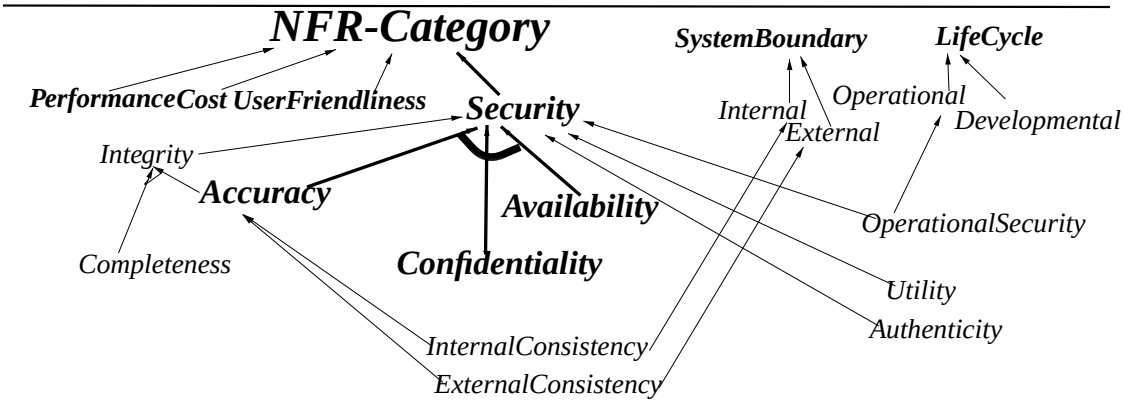
NFR framework ⟶ Systematic Process

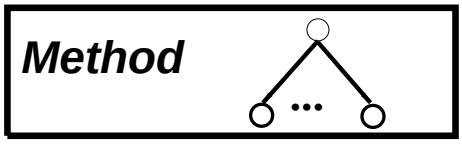


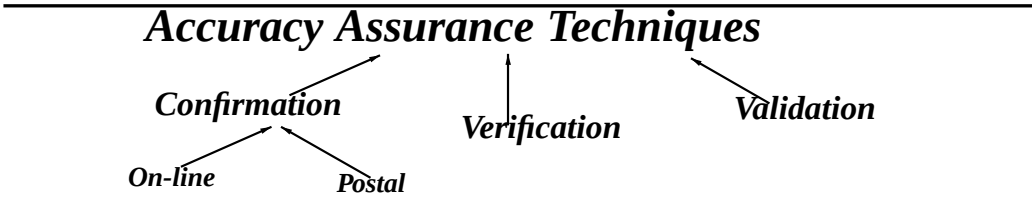
NFR framework ⟶ Systematic Process

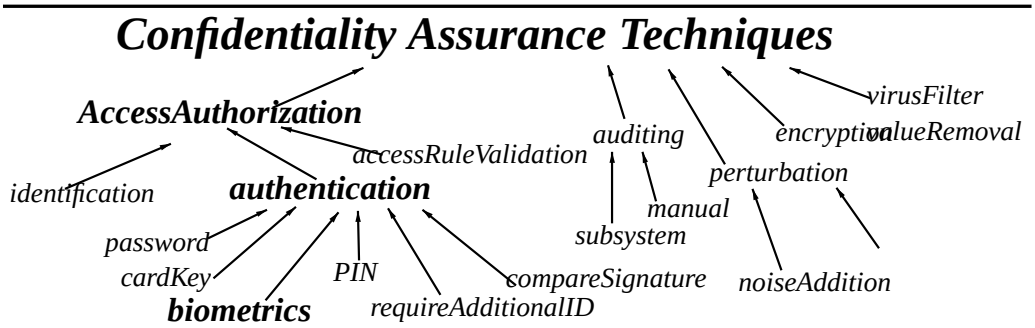
NFR framework ___ Systematic Process



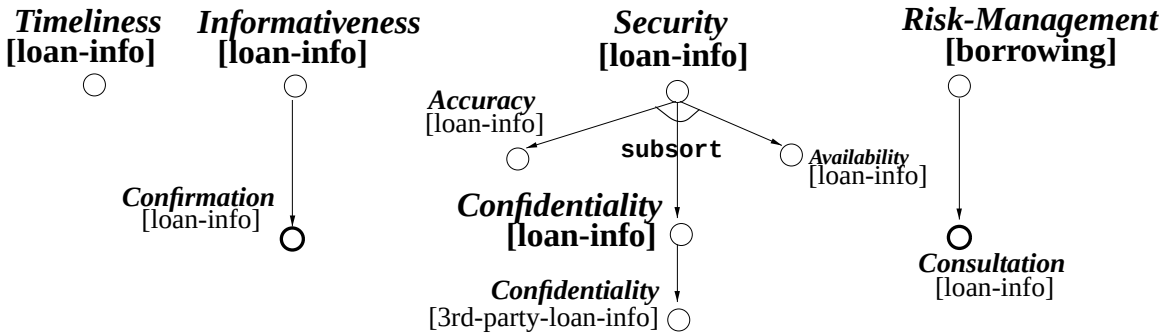




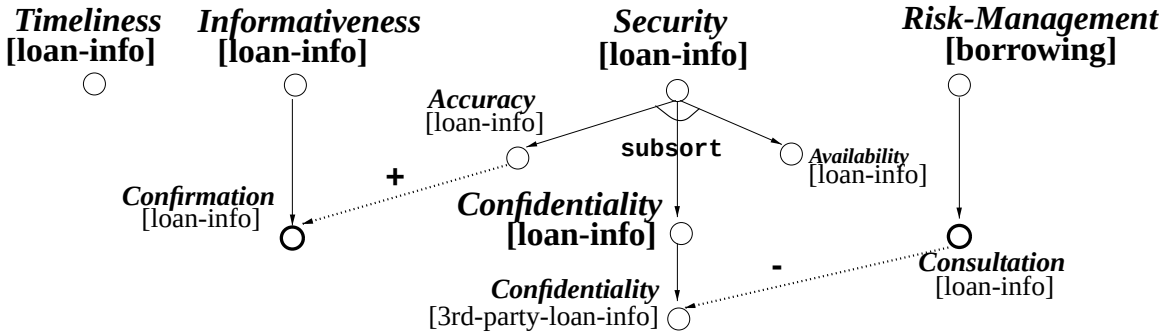




NFR framework ⟶ Systematic Process



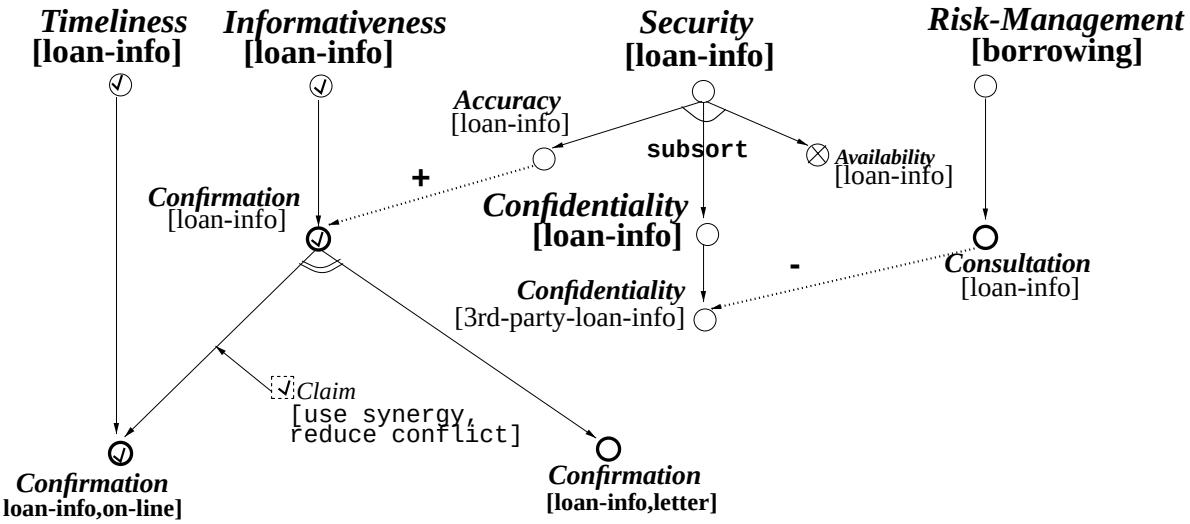
NFR framework ⟶ Systematic Process

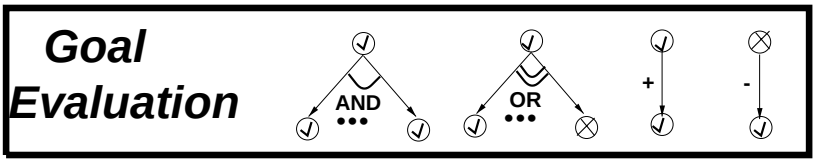


NFRGoal \ Technique	*Confirmation*	*Consultation*	*biometrics*
Accuracy	+	+	+
Confidentiality	+	-	+
UserFriendliness		++	- -
Response Time			- -
Risk-Management	+	+	

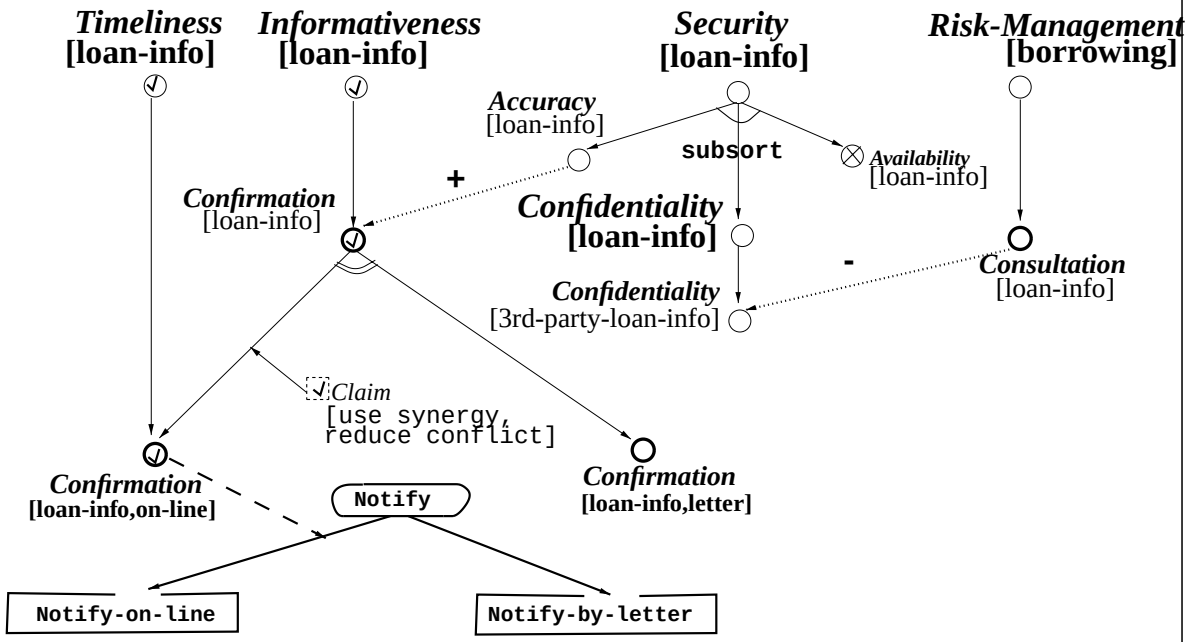
Correlation Rules

NFR framework ⟶ Systematic Process





NFR framework ⟶ Systematic Process

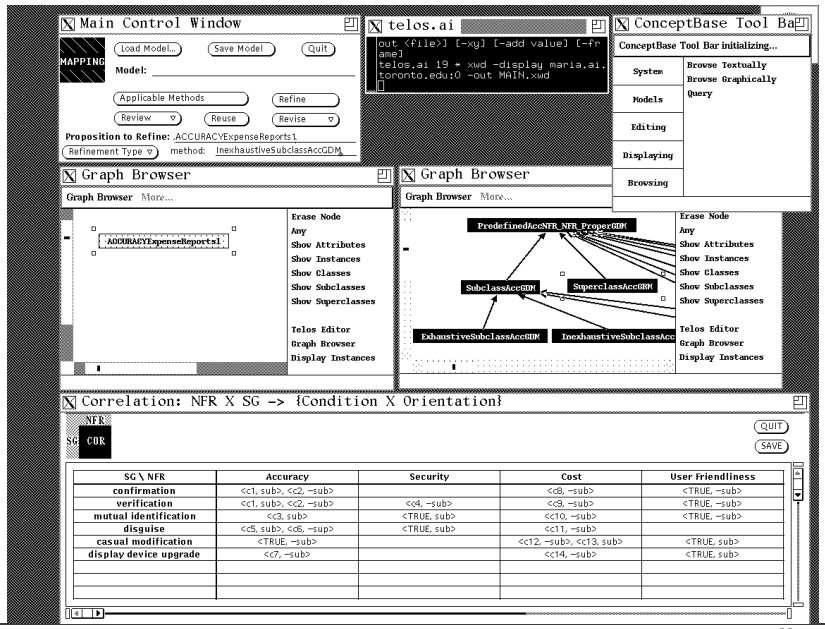


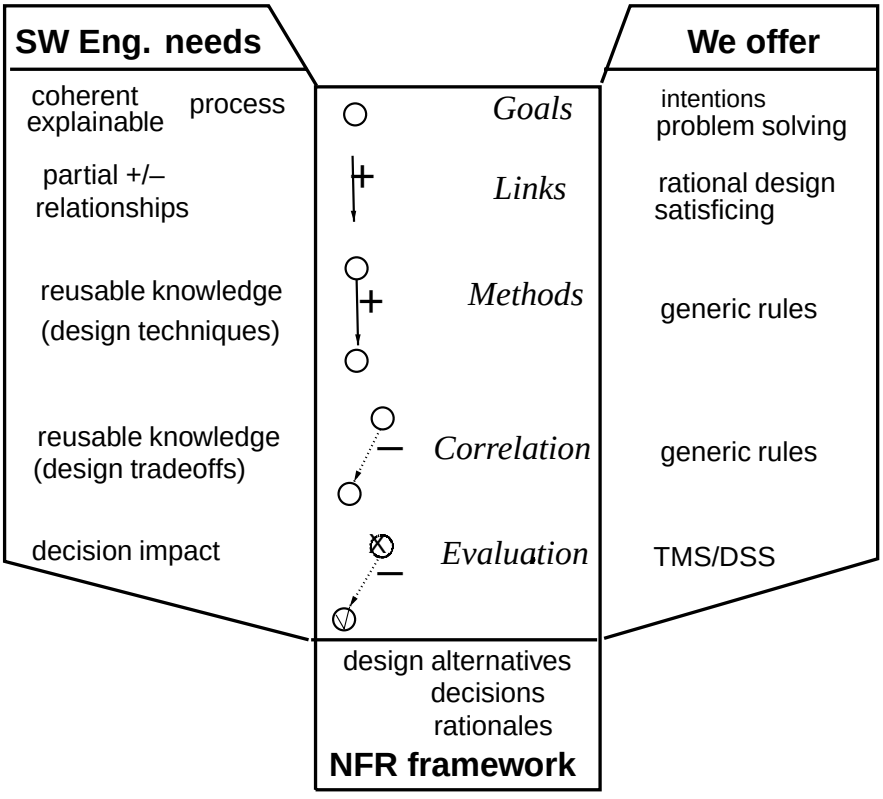
Prototype NFR Assistant

systematic approach $\Longrightarrow$ *assist development process*

Functionality: Goal Structure, Methods, Correlation Rules, Labels

Interface: Main Control, Graph Browser, Correlation Table, Text-editor





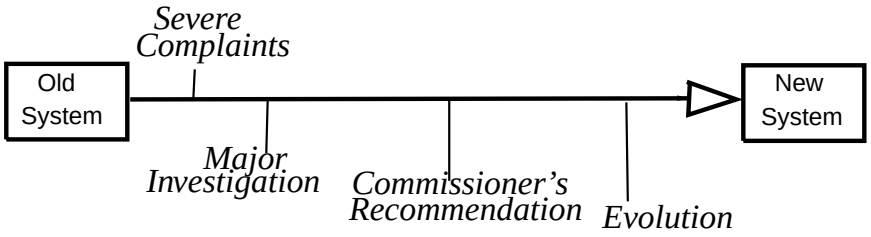
Three Small Applications

1. **Health Insurance**

2. **Credit Card**

3. **Government Administration**

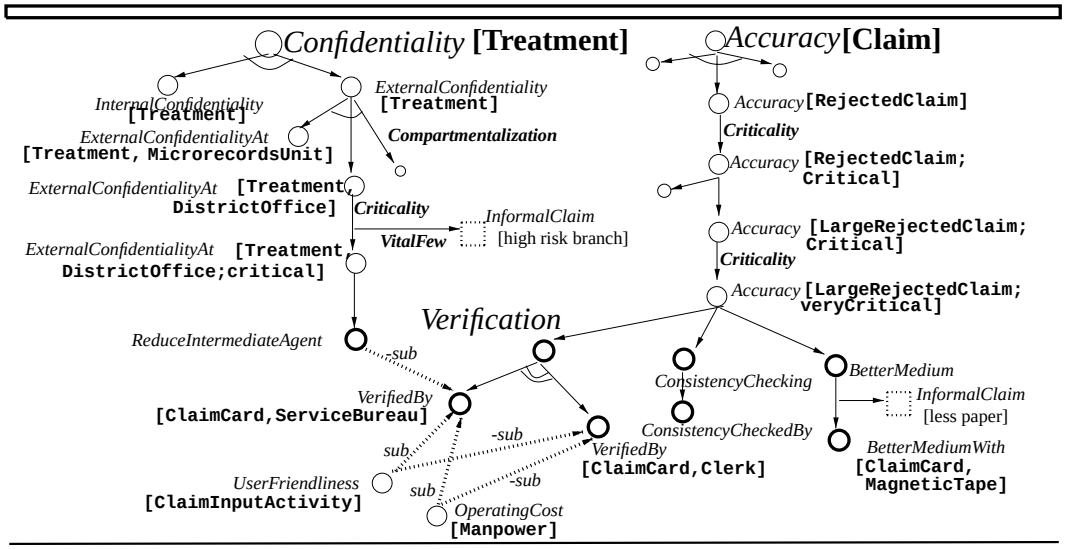
1. Health Insurance System



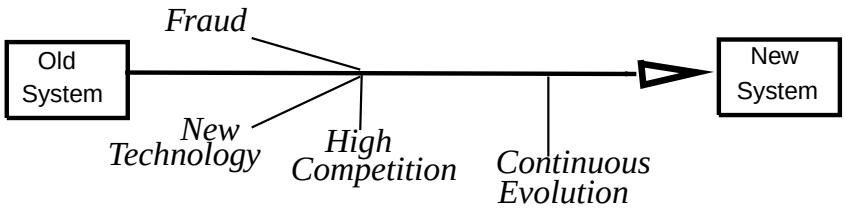
Method	
subsort (implicit)	system boundary, compartments
criticality	control point, discretionary
for accuracy	consistency-check, verify, better medium
for confidentiality	password, encryption (↑ external)
supporting documents	on-site reviews, questionnaires
	for why satisfactory

Correlation	
synergy	√ √
conflict	√ √

1. Health Insurance System



2. Credit Card System



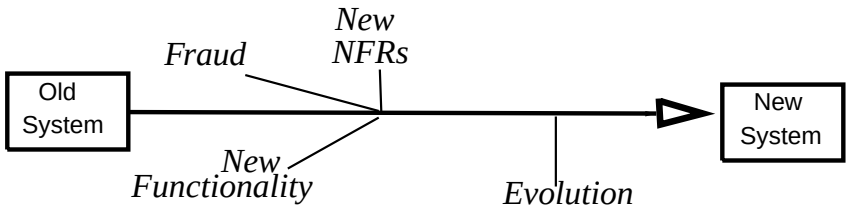
Methods

subsort, criticality (implicit)	√
for accuracy	reduce-trans-time, allow-direct-access
for confidentiality	access authorization (?), bulletin board
supporting documents	reviews & statistics, market survey (?)
	for synergy justififcation

Correlation

synergy	√ √ √
conflict	implicit

3. Government Administration System

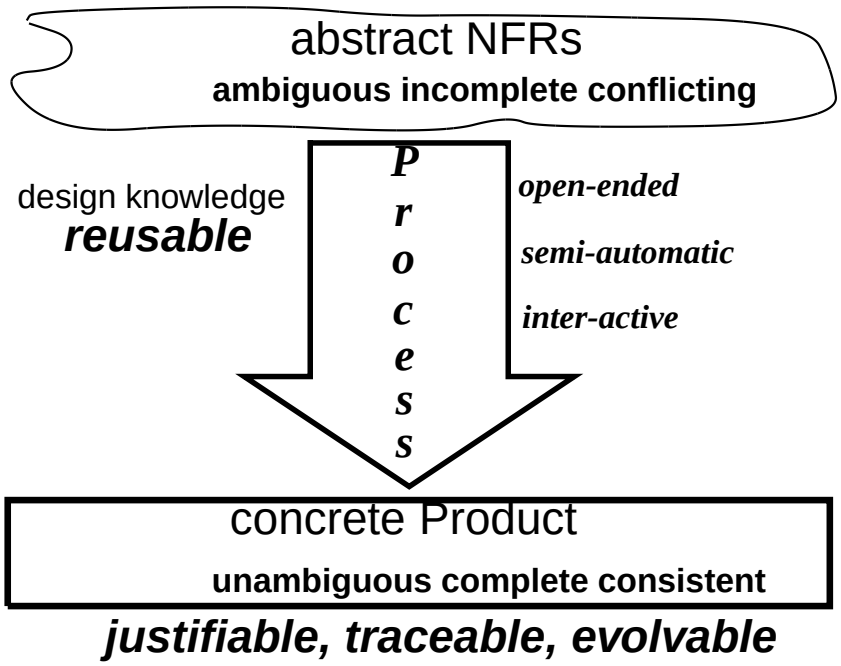


Methods	*subsort* (implicit)	$\surd$, access group
	criticality	implicit
	for accuracy	consistency checking
	for confidentiality	password
	supporting documents	ministry–standards
		for potential conflict
Correlation	*synergy*	implicit
	conflict	implicit

Some Observations

- Need to capture key concepts
  ←– met by framework components
- Framework helps reduce faults
  (ambiguities, omissions, conflicts, redundancies)
- Goal structure helps justify, trace, evolve
- Tool offers methods for vocabulary & subject matter;
  Partial automation reduces errors.

Quality Product — *through NFR framework*



In the Banking World ...

Can we build quality into software?

NOW A *systematic* approach to dealing with
Quality Requirements (Non-Functional Requirements — NFRs)
during software development

Status

Initial Research

- NFR framework
- Application to Accuracy and Security Requirements
- Tool Support, via NFR Assistant
- Small Application to 3 Real Information Systems

Other Applications

- Performance Engineering
- Organizational Modelling
- Project Risk Management
- Requirements Engineering

Future Directions

Improvements to NFR Assistant

- Performance
- User-friendliness
- Extensibility

Extensions to NFR Framework

- Formal Semantics
- Quantitative + Qualitative Reasoning
- Control Structure

Applications

- Life-size Information Systems
- Other Industrial Systems