CONCEPTUAL DESIGN FOR SOFTWARE PRODUCTS: SERVICE REQUEST PORTAL

Tyler Munger

Subhas Desa

Real World Problem at Cisco Systems

- Smart Call Home (SCH) is a component of Cisco Smart Services that offers proactive diagnostics and monitoring on network devices (routers, switches, etc.)
- The retrieval and extraction of intellectual capital (IC) from customer Service Requests is crucial to developing rules for SCH products
  - IC retrieval refers to basic keyword search for the purposes of locating relevant Service Requests
  - IC extraction refers to the more complex process of using the retrieved information for the development and management of SCH intellectual capital
- Currently the IC retrieval/extraction is time consuming and not easily scalable to new products

See paper (3)

Service Request Portal

- The Service Request Portal (SRP) is a software product we have developed that allows Cisco engineers to rapidly identify and extract IC from customer Service Requests
- The SRP performs two major functions:
  1. Reduces the **number** of Service Requests engineers have to read (Retrieval)
  2. Reduces the **amount** of Service Request content engineers have to read (Extraction)

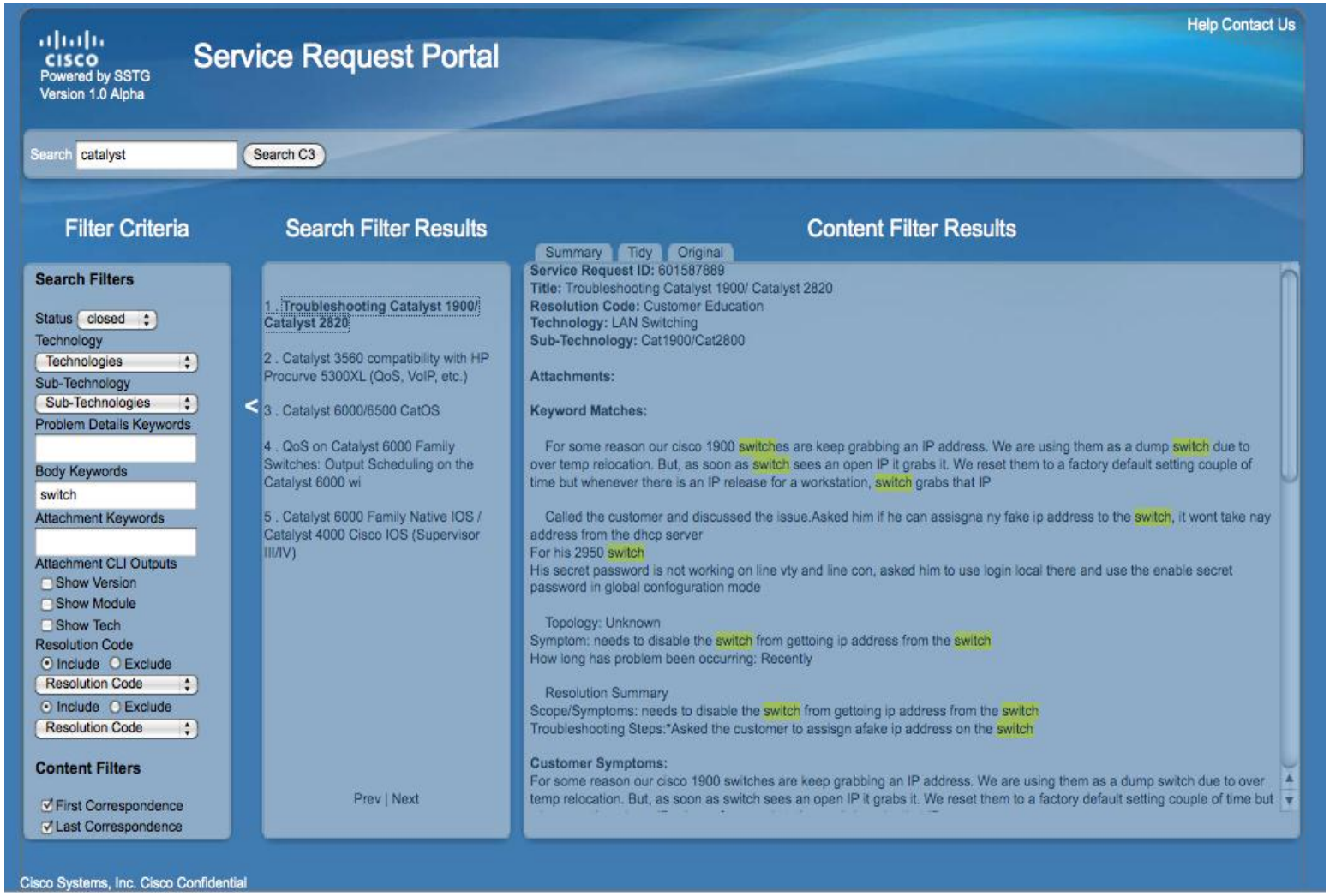
Service Request
603877653

Example Summary of
Service Request
603877653

SRP was developed in collaboration with Danny Core (UCSC) and Sri Ramachandran (Cisco)

See paper (41)

SRP: User Interface



See paper (41)

SRP: Results

- The SRP significantly improved the engineer's productivity when working with Service Requests
  - 60% improvement when retrieving IC
  - 30% improvement when extracting IC

- Initial user feedback:
  - "The portal looks just amazing. Our team will be using the portal for Syslogs IC creation"
  - "The tool helps to find the information easily when compared to C3 TOPIC search"
  - "It is useful to extract the information quickly"

See paper (42)

Conceptual Design for Software Products

- Central to the success of the SRP was the use of a Conceptual Design “front-end” that was applied before designing and implementing the software product
- Conceptual Design process:
  1. Knowledge Capture (Product Intent)
  2. House of Quality
  3. Due Diligence (Reverse Engineering / Market Research)
  4. Function Structure
  5. Morphological Matrix
  6. Selection Criteria
  7. Utility Function

See paper (11)

Step 1: Knowledge Capture (Product Intent)

- In order to determine what the product would need to do we developed a model of the users' current work process using interviews and focus groups
- The model was then used to identify areas of the work process that would benefit from software automation
  - Engineers have to manually examine the Service Request fields in order to determine if it is relevant
  - The relevant parts of the Service Request must be manually within extracted from many pages
  - Engineers must download Service Request attachments to their desktop before they could open them
- It was determined that the overall intent of the product would be: "Facilitate searching for relevant Service Requests and extracting problem-solution pairs"

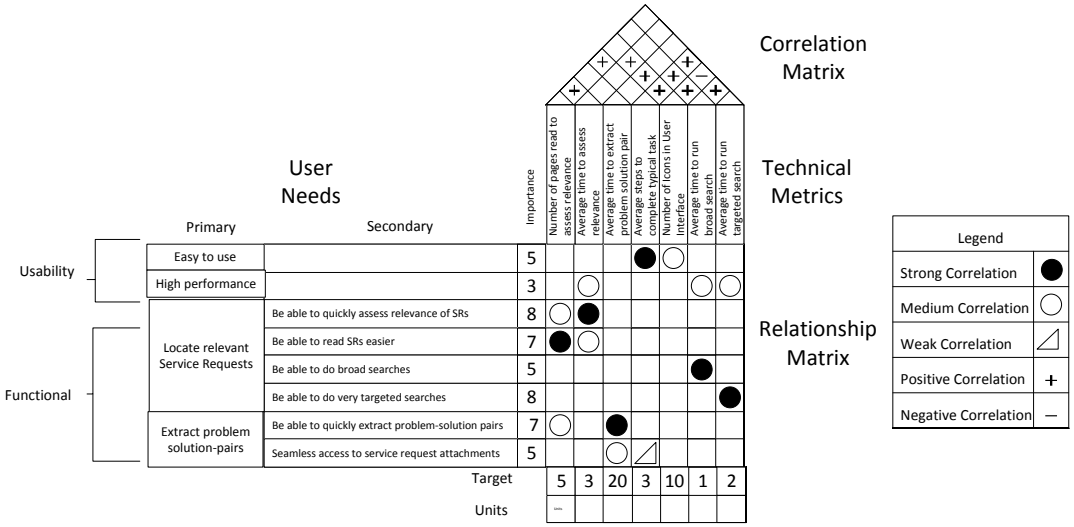
Engineers Work Process

1. Run a keyword search using service request database search engine to locate Service requests
2. Locate relevant Service requests in the search results
   a) Open the service request using the service request Viewer
   b) Examine service request fields (technology, sub-technology)
   c) Locate and read the problem description
   d) Locate and read resolution summary
   e) Locate and read first and last correspondence
3. Extract problem solution pairings from relevant Service requests
   a) Re-examine the problem description, resolution summary, and first and last correspondence in more detail
   b) Read case notes if necessary
   c) Download and read attachments if necessary
4. Use the problem-solution pairs to create smart network products

See paper (12)

Step 2: House of Quality

- The House of Quality was used to identify the user needs for the SRP and relate them to the Technical Metrics for testing the SRP during software development
- We focused on two types of user needs for the product:
  - **Usability needs**: characteristics that the product must embody, such as “easy to use”
  - **Functional needs**: functionality that the product will need to provide to the user, such as “be able to read Service Requests easier”
- The Technical Metrics were based on the engineers’ current work process



See paper (19)

Step 3: Due Diligence (Reverse Engineering/Market Research)

- A due diligence survey was performed in order to ensure that we leveraged existing work inside and outside of Cisco:
  - Surveyed past and present projects inside Cisco to understand the landscape
  - Comprehensive research of commercially available tools
- We determined, based on the survey, that the desired product did not exist either inside Cisco or as a commercially available tool. Therefore we would develop a new software product (the SRP) to address the user needs

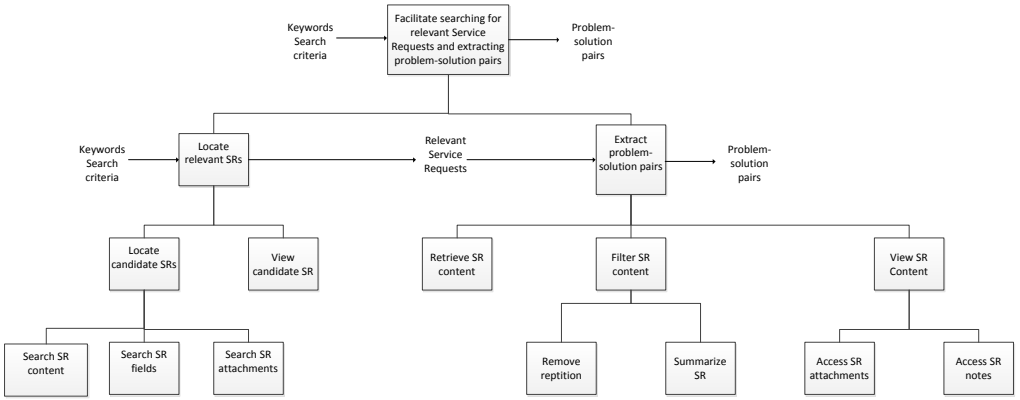
Cisco Projects	Group	Product Summary
Best Practices	STS	Rule Based framework for creating best practice suggestions from data collected by CNC
	AMG	Tool to automate the detection and correction of errors within device configurations
Product Adoption/Usage	AMG	•Tool to extract general product information from the CLI output contained in C3 attachments. •This information is used by Business Units and Marketing to determine product usage/adoption
Topic Enhancement	TS Web	• Migratation of TOPIC from the current Google system to the Microsoft FAST system. •This system will offer a number of improvements including direct access to the search data using an API

Software Tools	Products (Vendor)	Summary	Advantages	Disadvantages
Statistical Analysis	Enterprise Miner (SAS) Clementine (SPSS)	Toolsets for pattern and trend discovery in semi-structured data	Powerful and scalable Some of these tools such as SAS are already in use at Cisco	Expensive Difficult to use
Information Extraction/ Retrieval	Lemur (Open Source) GATE (Open Source)	Toolsets for automatically extracting structured information from unstructured data	Access to source code allows these tools to be modified to SCH's needs Low Cost (in dollars)	Lacking the documentation and user interface of commercial tools
Database Mining	Oracle Database Mining (Oracle)	Toolsets for trend and pattern discovery directly inside the database	Excellent performance/scalability Easier deployment because of direct database integration	Requires database access Not necessarily as mature as the specialized statistical analysis tools such as SAS and SPSS

Not in paper

Step 4: Function Structure

- The Function Structure was used determine the to connect the user needs with the functional specifications for the SRP
- Each SRP function is related to a user need in the House of Quality
  - The function “Locate relevant SRs is derived from the primary user need “Locate relevant SRs”.
  - The “Search SR content” is related to the user need “Be able to do broad searches”.
  - “Search SR fields” and “search SR attachments” are related to the secondary user need “be able to do targeted searches”



See paper (20)

Step 5: Morphological Matrix

- The Morphological Matrix was used to capture feasible solution principles (realizations) for the SRP's functional specification
- We tried to capture a range of solution principles for each sub-function from simple to very complex. For example, consider the solution principles for the sub-function "remove repeated content":
  - Low complexity: Strict hash based deduplication
  - Medium complexity: Fuzzy hash based deduplication
  - Very complex : Use a classifier to detect duplicate information

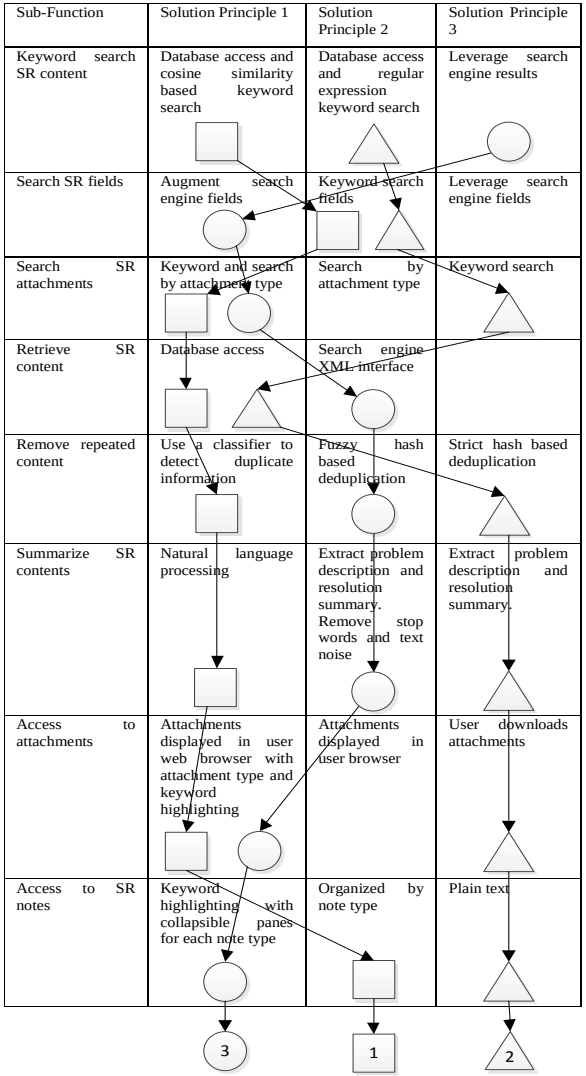
Sub-Function	Solution Principle 1	Solution Principle 2	Solution Principle 3
Keyword search SR content	Database access and cosine similarity based keyword search	Database access and regular expression keyword search	Leverage search engine results
Search SR fields	Augment search engine fields	Keyword search fields	Leverage search engine fields
Search SR attachments	Keyword and search by attachment type	Search by attachment type	Keyword search
Retrieve SR content	Database access	Search engine XML interface	
Remove repeated content	Use a classifier to detect duplicate information	Fuzzy hash based deduplication	Strict hash based deduplication
Summarize SR contents	Natural language processing	Extract problem description and resolution summary. Remove stop words and text noise	Extract problem description and resolution summary.
Access to attachments	Attachments displayed in user web browser with attachment type and keyword highlighting	Attachments displayed in user browser	User downloads attachments
Access to SR notes	Keyword highlighting with collapsible panes for each note type	Organized by note type	Plain text

See paper (22)

Step 6: Alternative Design Concepts

- Three different alternative design concepts for the SRP:

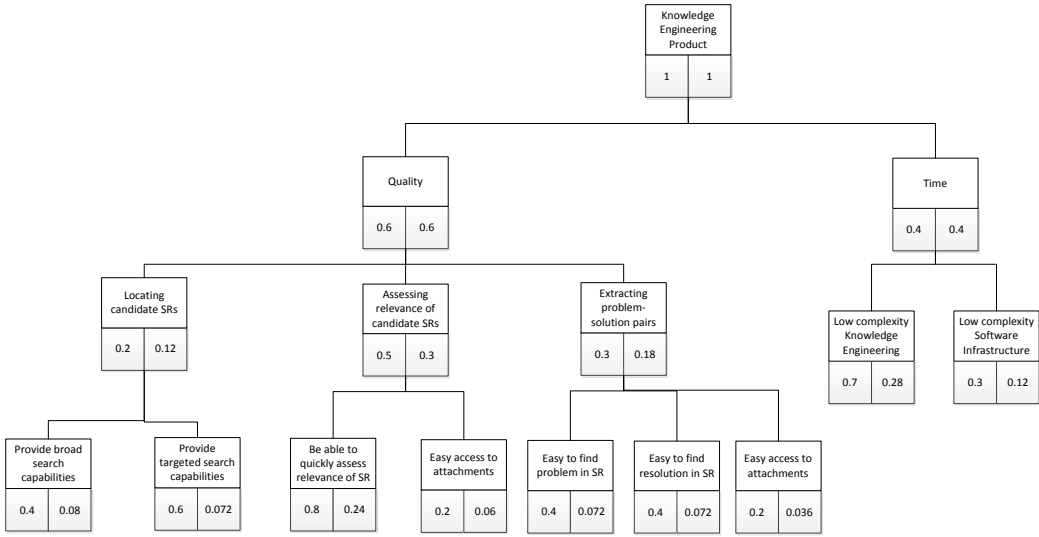
1. “State of the Art” approach: use complex solution principles such as natural language processing
2. “Hybrid” approach: combine existing Cisco software infrastructure with the new functionality desired by the users
3. “Quick and Dirty” approach: use very simple solution principles such as regular expression matching



See paper (24)

Step 7: Selection Criteria (Deriving the Utility Function)

- The Selection Criteria captured the development objectives for the SRP and allowed us to derive a Utility Function for evaluating design concepts with respect to these objectives
- There were two overall development objectives for the SRP:
  - **Quality**: maximize the user value. This corresponded to the user needs specified in the House of Quality
  - **Time**: minimize the development time for the SRP. This corresponded to the complexity of the solution principles (e.g. how long they would take to implement)



See paper (26)

Step 8: Concept Selection Using the Utility Function

- The “Hybrid” design concept (3) was the highest utility design. It offered similar user value to the “State of Art” design concept (1) and was significantly less complex because a large part of the infrastructure was already in place

Selection Criteria	Relative Weight	Design Concept 1		Design Concept 2		Design Concept 3	
		Score	Utility	Score	Utility	Score	Utility
Provide broad search capabilities	0.048	8	0.384	4	0.192	7	0.336
Provide targeted search capabilities	0.072	6	0.432	3	0.216	7	0.504
Be able to quickly assess the relevance of the SR	0.24	7	1.68	4	0.96	6	1.44
Easy access to attachments	0.044	4	0.176	2	0.088	7	0.308
Easy to find problem in SR	0.072	7	0.504	3	0.216	5	0.36
Easy to find resolution in SR	0.072	7	0.504	3	0.216	5	0.36
Complexity of Knowledge Engineering	0.28	3	0.84	8	2.24	6	1.68
Complexity Software Infrastructure	0.12	3	0.36	4	0.48	7	0.84
Cumulative Utility			4.88		4.61		5.82

See paper (28)

Step 9: Software Engineering (Manufacturing)

- A typical Software Engineering process is consists of two distinct workflows:
  1. Software Design (Use Case Diagrams, Component Diagrams, Class Diagrams)
  2. Software Development (Planning, Coding, Testing)
- We followed conventional Software Engineering methods with the exception that we radically altered the drivers that feed into the design and development processes
- Using the results of the Product Design phase ensured that the Software Engineering phase was sufficiently linked to the user needs for the product

See paper (31)

Software Engineering: Product Design Drivers

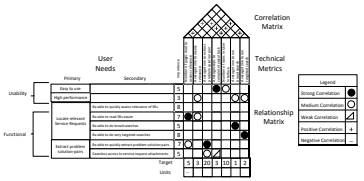
HOQ

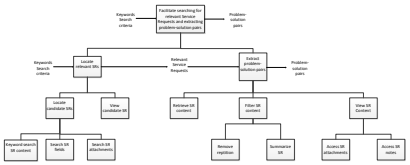
FS

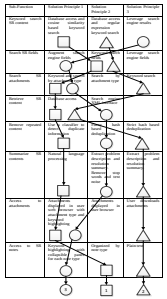
MM

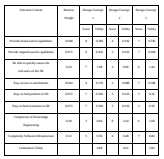
SC

UF







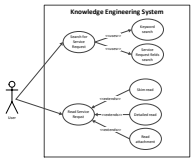


sub-functions

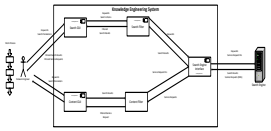
sub-functions, solution principles

technical metrics, user needs

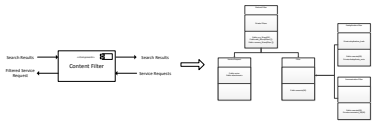
selection criteria



Use Case Diagram



Component Diagram



Class Diagram

Design

Planning

Testing

Development

See paper (31-40)

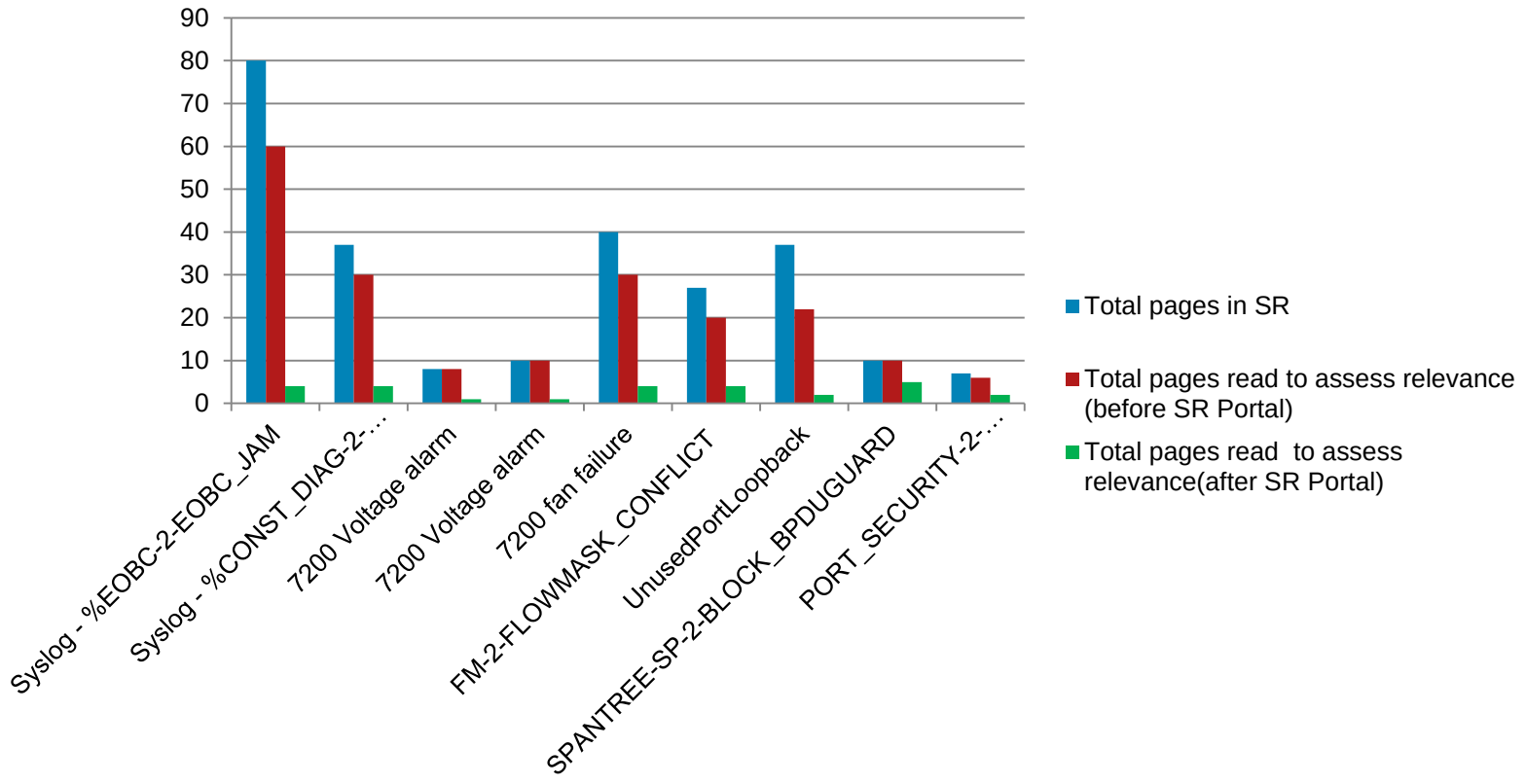
SRP: Implementation

- Software Infrastructure
  - Ruby on Rails/MySQL backend
  - HTML/Javascript user interface
- Retrieval features
  - Search for any combination of keywords or phrases within the body of the Service Request and text attachments
  - Include or exclude certain resolution codes
  - Search within any technology and sub-technology
- Extraction features
  - Grouping the bugs referenced in the Service Request summary and opening them from the portal
  - Opening text attachments from within the portal
  - Highlighting text around specified keywords in Service Request body in the summary
  - Present the first and last correspondence in Service Request summary
  - Highlighting text around specified keywords in Service Request attachment when opened
  - Tidy version gives nice color-coded logical flow of the Service Request and removes repetition

See paper (41)

SRP: Results-1

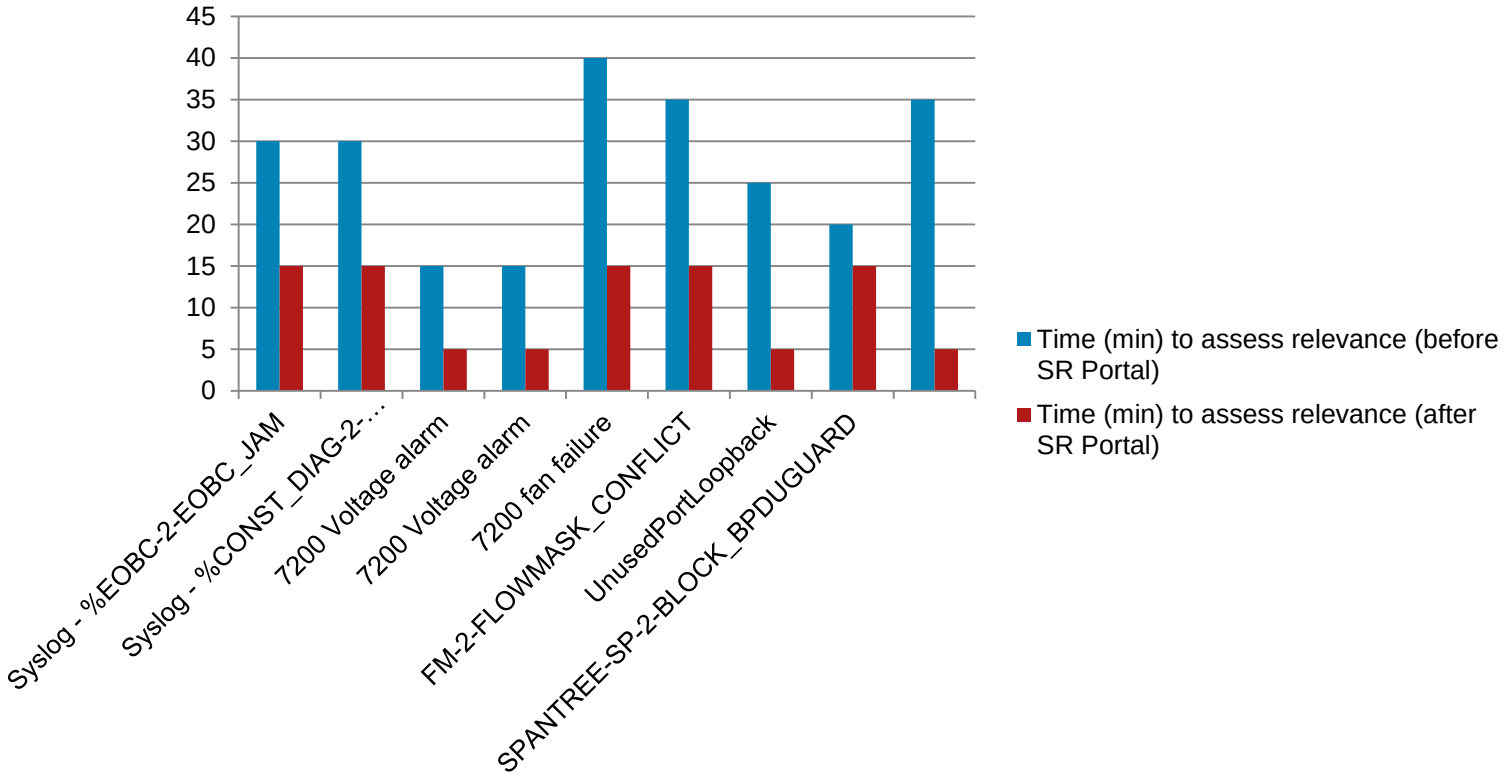
- Average reduction of 85% (20+ to 3) in pages read per Service Request when establishing relevance



See paper (45)

SRP: Results-2

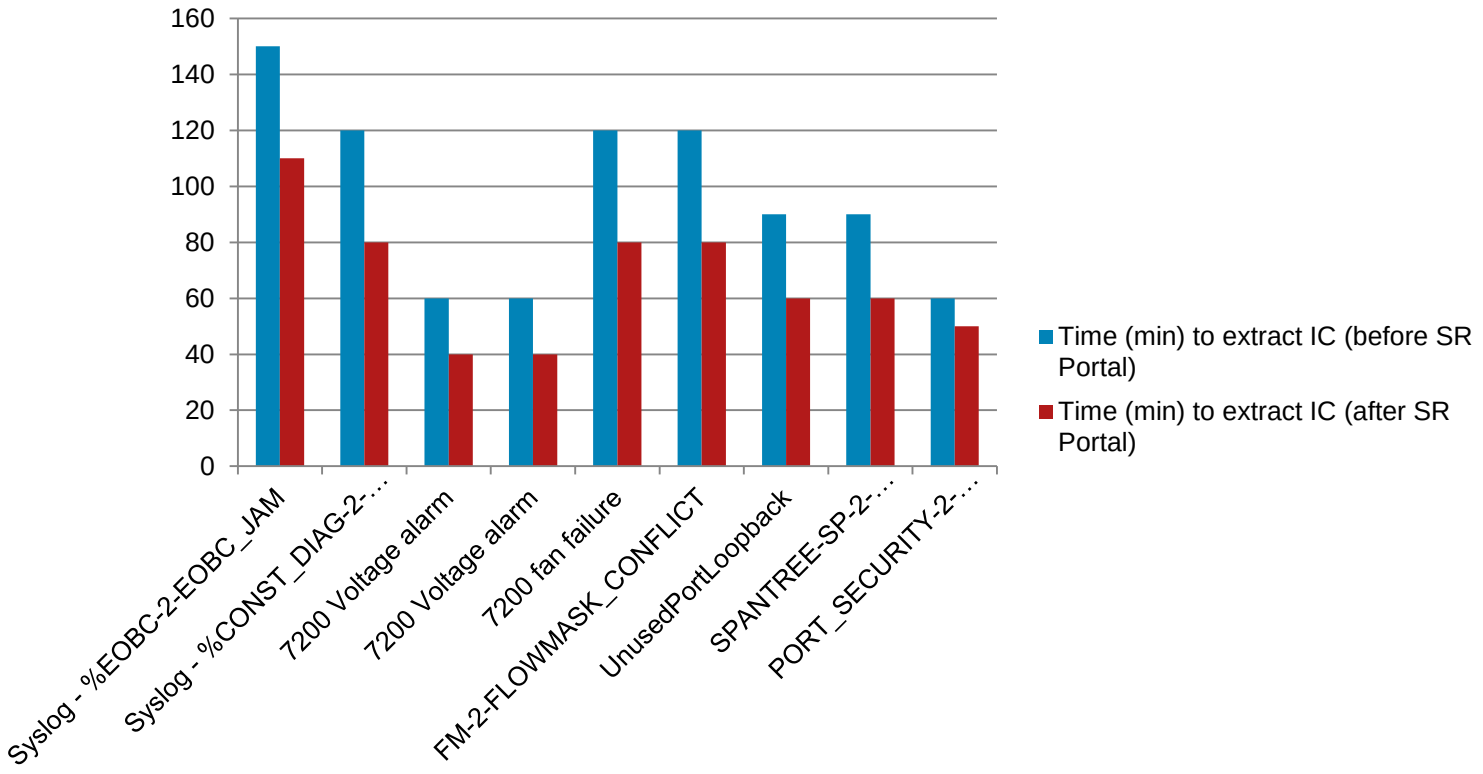
- Average time savings of 60% (15 minutes) per Service Request when establishing relevance



See paper (45)

SRP: Results-3

- Average time savings of 30% (30 minutes) per Service Request when extracting IC



See paper (46)

Conclusions

- Developing high quality software products at low cost requires:
  - Relating the user to the system
  - Thinking about the system functionally
  - Exploring different function realizations
  - Selecting the best “mix” of realizations based on the user needs and cost constraints
- Conventional Software Engineering does not provide structured methods for these activities
- Product Design methods and techniques can be used as a supplemental “front-end” to conventional Software Engineering methods in order to increase the quality and cost effectiveness of software products

See paper (46)