I. Goals

- To develop a deep understanding of what architects do
  - what they think architecture is
  - how they think about it
  - how they relate it to general requirements, detailed design and implementation
  - how they go about creating and evolving it
- To capture the meaning of software architecture in practice

II. Privacy issues

- Anonymity (of both architects and company) is guaranteed, unless explicit permission is provided by the interviewee and/or his/her company
- Access to the recording data is strictly limited to researchers involved in the project and no one else.
- Company confidentiality will be maintained on architecting process issues if requested.
- Company confidentiality will be maintained on IP issues, unless explicit permission is provided by the interviewee and/or his/her company

III. Questionnaire

- Note that the questions below may have overlaps, please feel free to repeat your answer if appropriate.
- The order of the questions can be altered according to the architect’s preference.
- Questions may also be skipped.
A. Problem Domain

A1. Describe the problem domain(s) that you have been working in.
   - General domain
   - Specific domain examples

A2. What are the characteristics in each domain?
   - Mature vs. immature
     - Mature – established business process; well defined theory, processing and expected behaviors; well understood objects
   - Stable vs. unstable
     - Degree or constancy of change during and after development
   - Physical vs. intellectual
   - Level of automatic
     - Automatic vs. manual
     - Sophistication and quality of the automation

A3. What effects the domain have on requirements and architecture?

A4. Coverage of domain
   - General purpose vs. specific purpose
   - General users/uses vs. specific users/uses

B. Overview of Software Architecture

This section is to capture the architect’s general understanding of software architecture and its meaning and significance to the system.

B1. What do you consider to be the critical characteristics of software architecture?
   - What is the essence of software architecture?
   - How many levels of architecture are there? How detailed should it be? Should it be prescriptive or descriptive?
   - What would the architectural representation include?
   - How do you communicate an architecture to the stakeholders?
B2. What are the critical aspects and characteristics of a great/superb software architect?
   - Politically
   - Technically
   - Other aspects?

B3. What drives the architecture?
   - Why do you need an architecture?
   - What are the driving forces in its creation?
     - Is it the non-functional requirements,
     - the functional requirements,
     - the particular blend/mix of both,
     - the organization's culture,
     - or business needs?

B4. What is the meaning of architecture in your opinion?
   - Not the definition but the meaning of architecture
   - What does an architecture represent?
   - Is it merely used for communication and comprehension, or something more?
   - How does architecture manifest itself in the final system since it is not visible in the code?

B5. Given all of the above discussions, how do you view software architecture?
   - Is an architecture a technical need, or business need?
   - How do you see them being different?

C. Requirements and Architecture

This section helps us to understand the architect’s opinion about the relation between the requirements and the architecture.

C1. How do you view requirements?
   - How do you distinguish functional and non-functional requirements?
   - How do you deal with inconsistent requirements?
   - What form do your requirements take?
C2. How do you predict and deal with requirement variance?
   • Solving the right problem?
   • Do the requirements cover a sufficient area in the problem space?
   • What do you do if the supplied requirements are
     o Whether it is too narrowly defined,
     o too open and unspecific, or
     o with hidden assumptions.
   • How to anticipate variance and future requirements?
   • What environment factors help you to determine the variance?
   • How do you handle ambiguity and incompleteness in the requirements?

C3. How do you transform the functional requirements into an architecture?
   • Do non-functional requirements play a role in this transformation?
   • Any examples?

C4. How do you handle the non-functional requirements?
   • Do you integrate these with the functional ones initially or after you have
     considered the functional ones and built a skeleton architecture?
   • How do the functional and non-functional aspects interplay in the design of an
     architecture?
   • Is there an ordering or a set of priorities for non-functional requirements?

C5. Do you have domain-specific standard or generic styles, structures, patterns,
   transformations, or techniques for non-functional requirements?
   • For example, there are standard techniques used in telephone switches - eg,
     watch dog timers which are used for certain kinds of reliability and fault
     diagnosis techniques.
   • Do you have preferred ones and why? Examples?

C6. How does the problem structure affect your architecture?
   • Do you think there is a relation between the problem structure and a good
     architecture? What would it be?
   • Are requirements sufficient to help you understand the problem structure?
   • Does it make sense to view doing requirements as a design effort?
   • What would you need before you can move towards that direction in terms of
     methods and tools?
C7. While designing the architecture, what do you look for in the requirements?
   - Do you try to reflect the unique characteristics of the problem in the architecture?
   - What are some examples?

C8. In designing an architecture, what do you do with requirements that you know will entail an overly complex or costly system?

C9. How do you evaluate the architectures?
   - Is there a formal evaluation process? How does it work?

D. Relating to Evolution

D1. How do new requirements affect the architecture after the system is built?

D2. How do you handle continuous requirements change?

D3. How does the architecture evolve during the system’s lifetime in response to changing requirements?
   - How do you deal with architectural drift or erosion?

D4. What measures do you take while designing the architecture to minimize the impact of change?
   - How do you do to identify and understand the various effects of requirements changes?

D5. How do you reuse an architecture?
   - How do you make an architecture reusable?
   - Are you usually concerned with reusability while designing?
   - Do you make use of product line architectures?
   - Do you find common parts that you can reuse?
E. Professional Background

To gain a basic understanding of the architect’s professional experience in architectural design and software development.

E1. Describe your overall professional architecting experience.
   • Duration, number of projects, size of projects, budget, success/failure etc.

E2. Describe the architectural (and/or requirements) aspects of one successful project that has left a profound impression on you.
   • What played a critical role in success/failure?

E3. Repeat A.2 for a failed or not so successful project.

E4. Have you had a successful project where you felt the architecture was not a good one?

E5. What about the reverse: unsuccessful with a good architecture?

F. Comments

F1. What do you think of the questions?
   • Are they relevant?
   • Do they help you to think differently of architecture compare to before?
   • Do you find this interview a useful exercise? Why and how?

F2. Do you have any recommended architects who might be interested in the interview?
   • Names and contact info.