by **Antony Tang**, RegPM, School of Information Technology, Swinburne University of Technology, Melbourne

Managing project risks with architecture modelling

IT PROJECT MANAGERS ALWAYS FACE UNCERTAINTIES WHEN PLANNING PROJECTS THAT USE NEW TECHNOLOGIES.

In these projects, risks or uncertainties often arise in critical areas such as technical feasibility, technology implications of requirements, and organisational capabilities.

As IT project managers, we prepare software development project plans that include budgets and schedules for stakeholders' approval.

But, accurately planning software development projects that involve new technologies can be difficult when the architecture risks are largely unknown.

To control these risks, technical feasibility studies or a pilot project should be undertaken before project planning commences. This step is often ignored.

While excitement or urgency pushes a new project forward, the risks of applying new technologies can be conveniently overlooked because the team is not aware of them.

Architecture modelling could help alleviate these problems.

Project risks

When the project plan is approved, neither the project manager nor the technical team are aware that there are usually gaps in the understanding of how the technology is to be applied. These gaps usually fall into a number of areas:

- integration issues with existing infrastructure
- limitations of the technology because of assumptions made by the technical team
- non-functional requirements such as performance, scalability, usability and security are not considered
- underestimation of the effort required to learn the technology
- the technology is not used properly due to the lack of experience
- features or sometimes bugs in the technology that require vendor or supplier support.

These technical issues may turn into major risks:

- cost – resources to investigate or solve the issue, purchase additional software or carry out additional development
- schedule – the schedule is delayed and its extent may not be known initially
- impact on requirements – business processes and requirements may be compromised and quality of services such as performance, reliability, etc. may not meet the required standards
- confidence level – any such issues often raise doubts among management and major stakeholders.

Architecture modelling

Architecture is defined by IEEE (IEEE 2000) as "the fundamental organisation of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution".

There are industry standards and frameworks such as the Zachman Framework (Sowa and Zachman 1992), ISO Reference Model for Open Distributed Processing (ISO/ITU-T 1997) and The Open Group Architecture Framework (The Open Group 2003).

These frameworks provide a structure to model architecture using views. A view is a perspective of a system with a particular focus. Commonly used views are:

- business view – model the functional and non-functional requirements and business rules
- information view – model the information and data requirements of the system
- computation view – model system functions, system interface and interoperability
- technology view – model the implementation on system platforms

Based on these views, a high level architecture model is developed at the beginning of a project to ensure its technical viability and consistency.

This design contains a group of inter-related views; a change of design in one view would have implications on others. For example, an integrated business workflow requirement (business view) may limit the design choice to a centralised database (information view) and centralised workflow engine (computation view) to support a nationwide operation. Hence, there are implications on user access, network traffic, performance and service availability (technology view).

The business view contains information on business rules, business requirements, business constraints, and strategies of the project.

The information view is the data and the information required to support the system, based on functional requirements. It also includes data interface and data transformation.

The computation view represents the architectural design of the software, such as components, classes and so on.

How architecture modelling works

Let's assume there is a new development for an online purchasing system. The system has a very high daily transaction rate and requires interface with a financial institution for payment processing. Using the architecture framework, the following key issues or assumptions should be addressed:

- business view – is there a clear definition of the business rules or business requirements? Are there obvious gaps or inconsistencies? Are there other interface requirements, such as an external enterprise resource planning system, to complete the business process? Are there defined non-functional requirements? Are there organisational standards the project must adhere to?
- information view – what information is required to support the system? Where does the information come from? Is there any requirement for data transformation or information transfer, and if so what is the interface specification? What would a high level logical data model look like?
- computation view – how is the new technology used in the system? Are there interface or compatibility issues with existing infrastructure? How would software components be modelled using the new technology? How would the developed components be deployed in a test and production environment? How do non-functional requirements influence design of the system?
- technology view – are there issues with meeting required security standards at the software, system and networking levels? What is the expected performance on the proposed platform? What is the required flexibility? What are the reliability requirements? What networking infrastructure will be used? Does the infrastructure and architecture model meet the required functional and non-functional requirements? Should tests or pilots be carried out to ensure the technology works as desired?

The results of these modelling activities are architecture models and specifications. System architects should work with project managers and stakeholders so major assumptions, constraints and tradeoffs of the system are agreed and captured. This process would reduce the risks of a project significantly and enable project managers to base their plan on accurate information and assumptions.

The technology view represents technical aspects of the system, such as hardware and software configuration, platforms, installation and configuration. There is often a misinterpretation that system architecture is represented by the technology or the software views. This is untrue because these views can only be modelled in the context of all the other views. All views are interrelated and a holistic architecture design approach is the only way to ensure the architecture is wholesome and there are no major design gaps.

The benefits

In a project using new technologies that are largely untested, architecture modelling should be performed as one of the initial and distinct phases in the development life cycle. Project managers would have more options to mitigate risks of the project if major issues are identified during architecture modelling.

The project manager could:

- negotiate with supplier or vendor for resolution
- reset expectations with stakeholders
- adjust the project plan accordingly
- depending on the nature of the risks, take steps to mitigate or avoid them
- make compromises and tradeoffs by renegotiating requirements or budget
- select another technical solution.

Ideally, finalisation of the project plan could be contingent on the result of the architecture activities.

The extra resources required to carry out architecture modelling would be offset by reduced resources in the design phase with the benefit of risk reduction.

Another benefit of architecture modelling is it would improve quality of delivery because a holistic design approach usually does not leave major gaps in satisfying requirements.

Other benefits of architecture modelling include early project buy-in and better stakeholder planning.

An architecture model could also be used to communicate with stakeholders on how the system would function to get early feedback and buy-in.

An architecture model is a blueprint of the structure of the system and it should identify major constraints, assumptions, limitations and tradeoffs. When they are assessed, they will provide a solid foundation for project budgeting and schedule estimation. PM

Email: atang@it.swin.edu.au

References

IEEE (2000), *IEEE Recommended Practice for Architecture Description of Software-Intensive System (IEEE Std 1471-2000)*, IEEE Computer Society.

ISO/ITU-T (1997), *Reference Model for Open Distributed Processing (ISO/ITU-T 10746 Part 1 – 4)*, Information Standards Organisation.

Sowa, J & Zachman, J (1992) Extending and formalising the framework for Information Systems Architecture, *IBM Systems Journal*, 31, 3.

The Open Group (2003) The Open Group Architecture Framework (ver 8.1 Enterprise Edition), URL http://www.opengroup.org/architecture/togaf/#download, Accessed 21 May 2004.