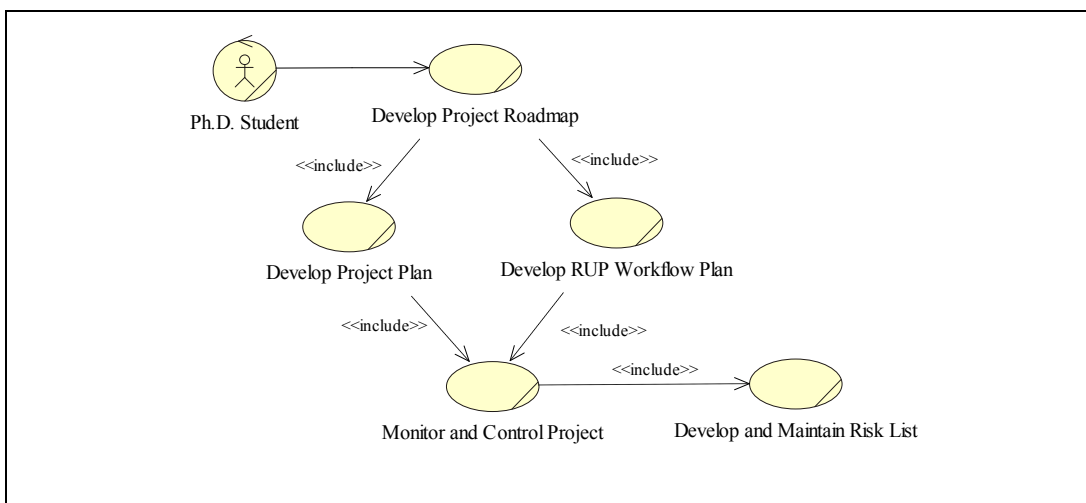


## F SACAPP ROADMAP

The SACAPP roadmap considered the vision of the PhD candidate for a new CAPP system through to the software prototype, and so, goals had to be defined, decisions that had to be explained, and, because a strong working discipline was needed, plans that had to be followed (Figures F.1, F.2, and Tables F.1 to F.7)). In addition, based on the best current software development practices (Reed, 2002), the UML was used to plan a system before it was built and provide a smooth transition between the business domain and the computer domain (Jacobson et al., 1999). Furthermore, because in the SACAPP life cycle reiteration of different phases occurred, changes had to be incorporated, errors had to be corrected, and the quality for both the development process and the resulting product had to be assessed, the RUP iterative, risk-driven, use-case, and business-focused software development process model was followed; tests were conducted to verify and validate that the built system was the right system that met its requirements; and then the results were assessed against similar process plans found out in industry (Figure F.3). Finally, the selection of object-oriented Java platform was in line with the UML object-oriented diagrams developed using Rational Rose, and synchronised with the code generated in JBuilder 7 Enterprise.



**Figure F.1** Project roadmap use-case main diagram

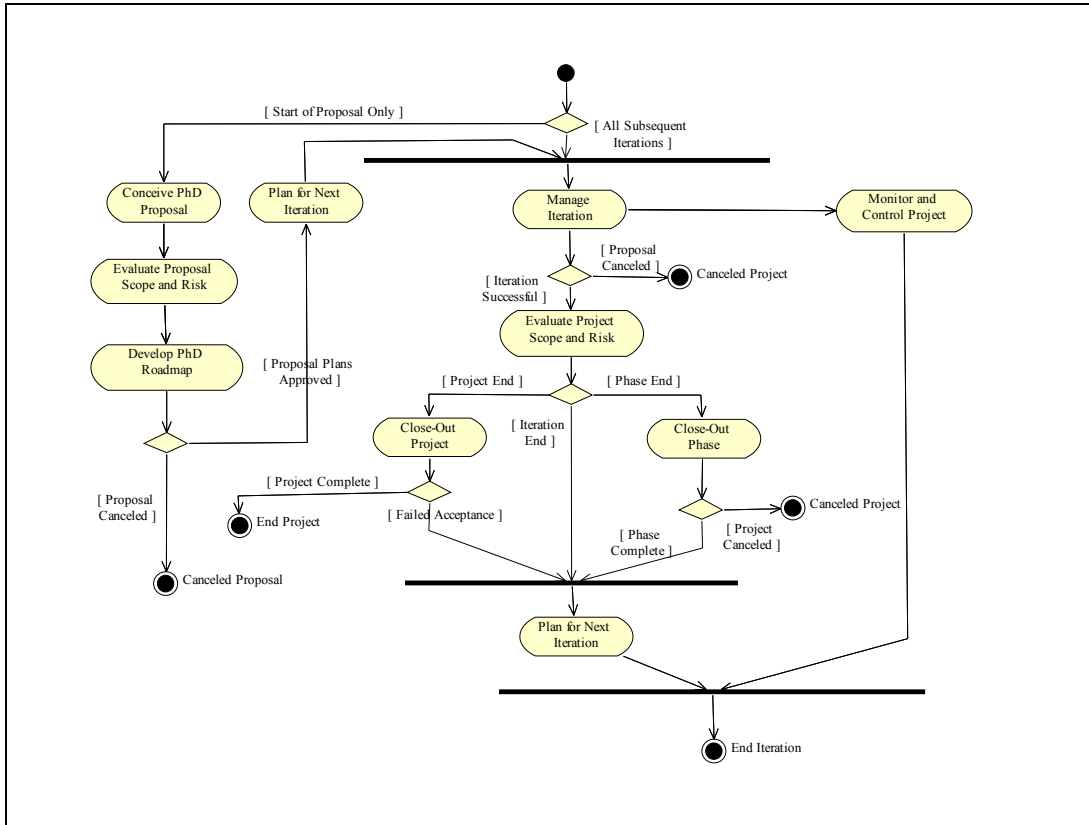


Figure F.2 Develop project roadmap activity diagram

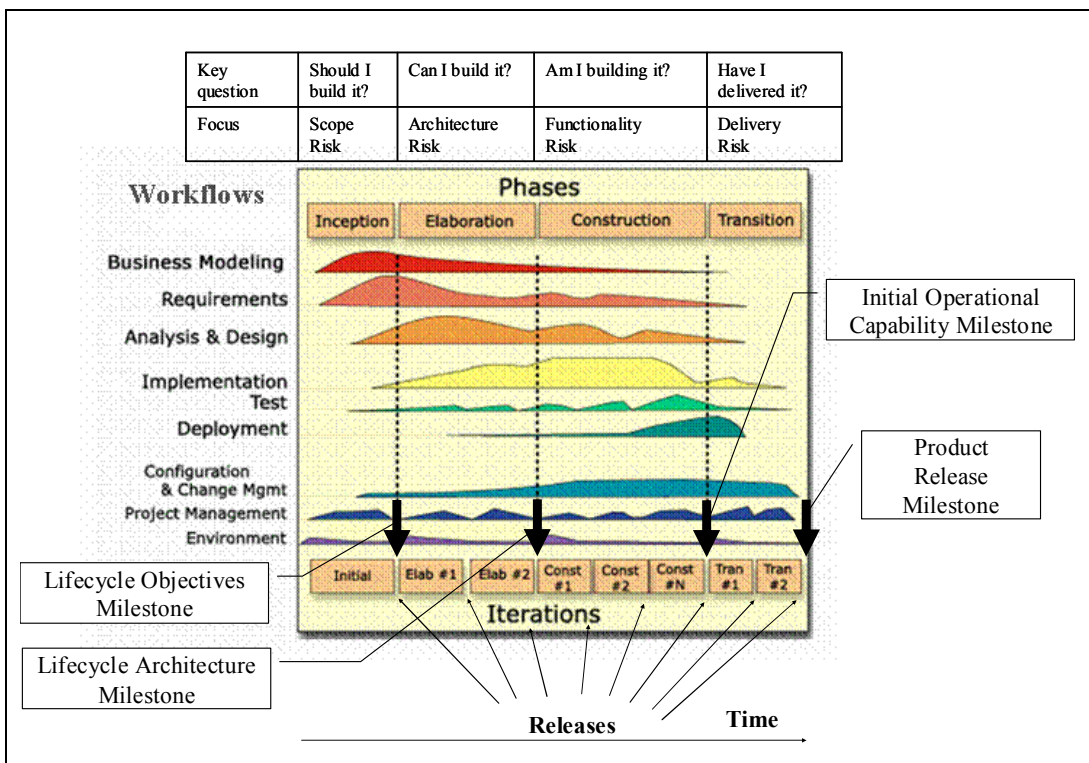


Figure F.3 SACAPP process model using RUP and its milestones

Table F.1 Project Plan

No	Task Name	Inception	Elaboration	Construction	Transition	Artifacts
1	Develop vision	Complete				Project Vision (Appendix G)
2	Capture common vocabulary	Initial	Complete			Appendix A, Glossary
3	Identify the features that the system must support	Initial	Update	Complete		Requirements plan and the List candidate requirements (Appendix G)
4	Identify, evaluate, and manage risks	Initial	Update	Update	Complete	Risk Management Plan (Table F.2), Risk List (Table F.3)
5	Develop use case	Initial	Update	Update	Complete	Use-Case Priority List
6	Develop product acceptance plan	Initial	Update	Update	Complete	QA Plan (Table F.6)
7	Schedule, monitor, and control project	Initial	Update	Update	Complete	
8	Plan and assess phases and iterations	Initial	Update	Update	Complete	Environment plan (Table F.7)
9	Handle exceptions and problems	Initial	Update	Update	Complete	Configuration and change management plan (Table F.4)
10	Project review authority (PRA)	Initial	Update	Update	Complete	
11	Close-out project	-	-	-	Complete	

**Table F.2** Risk Management Plan

No	Task Name	Inception	Elaboration	Construction	Transition	Artifacts
1	Evaluate/Re-evaluate project scope and risk	Initial	Update	Update	Complete	Risk List (Table F.3)
1.1	Identify the risk factors	Initial	Update	Update	Complete	Yes
1.2	Determine the risk exposure	Initial	Update	Update	Complete	Risk List (Table F.3)
1.3	Have all the critical risks identified?	Initial	Update	Update	Complete	Yes
1.4	Have the identified risks been mitigated or is there a plan for mitigating them?	Initial	Update	Update	Complete	Yes
1.5	Develop strategies to mitigate risks	Initial	Update	Update	Complete	Yes
1.6	Monitor risk factors and take action when needed	Initial	Update	Update	Complete	Yes
1.7	Update risk list	Initial	Update	Update	Complete	Yes

NOTE: Risk reduction was considered central to the project and necessary to be brought to the surface early and often (Vliet, 2000).

Table F.3 Risk List

No	Risk Description	Type	Probability 'p'	Impact 'I'	Risk Exposure ('p' times 'I')	Priority (critical, significant, or routine)
1	Unclear objectives	Organisational	2	10	20	Routine
		Strategy: Avoid it! Have clear objectives set-up by the PhD proposal.				
2	Not considering project risks	Organisational	2	10	20	Routine
		Strategy: Avoid it! Choose RUP iterative and risk-driven development strategy; perform risk management, and update the Risk List				
3	The project involves different or complex domains with a spread of application knowledge	Knowledge	3	10	30	Routine
		Strategy: Mitigate it! Build-up knowledge to better understand the relationship between different domains; Simplify complexity; Divide and conquer. Contingency: Re-assess objectives and project boundaries.				
4	Personnel shortfall, such as inexperience with the domain, tools or development techniques to be used.	Personal	3	10	30	Routine
		Strategy: Mitigate it! Carry out intensive training program. Contingency: Re-assess objectives and project boundaries.				
5	Unrealistic schedule	Organisational	3	9	27	Routine
		Strategy: Monitor it!				

		Contingency: Reschedule; Plan work during sabbatical.				
6	Unrealistic budget	Organisational	8	7	56	Significant
		Strategy: Monitor it!				
		Contingency: Use software tools and documentation in hand; Adapt to conditions.				
7	Wrong functionality, or not getting requirements right, or imperfect understanding of the customer needs, and insufficient domain knowledge	Organisational, Knowledge	1	10	10	Routine
		Strategy: Avoid it! Find what the users really want by an intensive industry interaction; Develop a comprehensive business model.				
8	Unstable, immature, unrealistic, or excessive requirements	Knowledge	4	10	40	Significant
		Strategy: Confine it or avoid it! Divide and conquer.				
		Contingency: Change requirements.				
9	Not finding a practical CAPP candidate architecture for problem solving that is resilient and can evolve gracefully during its lifetime.	Knowledge	7	10	70	Critical
		Strategy: Confine it! Simplify complexity; Divide and conquer. Build-up knowledge to better understand the relationship between different domains; Consider reliability, availability, reusability, system interface integrity, adaptability, and portability;				
		Contingency: Re-assess objectives and project boundaries; Change requirements.				
10	Use new technologies, such as computational techniques, not yet	Technology	7	10	70	Critical
		Strategy: Confine it! Limit it to only some use cases.				

	well developed	Contingency: Re-assess objectives and project boundaries; Change requirements.				
11	Not performing (speed, capacity, accuracy)	Technology	4	9	36	Significant
Strategy: Mitigate it! Contingency: Re-assess the techniques used; Change requirements.						
12	Wrong user interface	Knowledge	4	8	32	Significant
Strategy: Mitigate it! Contingency: Find what the users really want by an intensive industry interaction; Develop a comprehensive business model. Simplify complexity; Divide and conquer.						
13	Gold plating or ‘nice’ feature not asked for by the customer	Organisational	2	5	10	Routine
Strategy: Avoid it! Limit to necessary features.						
14	Not communicating properly with sales and design	Knowledge	7	10	70	Critical
Strategy: Confine it! Limit it to only some use cases. Contingency: Re-assess the techniques used; Change requirements.						
15	Not generating the correct process planning sequence of operations	Knowledge	7	10	70	Critical
Strategy: Confine it! Limit it to only some use cases. Contingency: Re-assess the techniques used; Change requirements.						

**Table F.4** Configuration and change management (CCM) plan

No	Task Name	Inception	Elaboration	Construction	Transition
1	Plan project configuration & change control: establish policies and change control processes	Complete			
2	Create and set-up project CCM environments	Complete			
3	Create, deliver and manage baseline (a complete set of agreed project documents) releases and its configuration items	Initial	Update	Update	Complete
4	Manage the submission, update, review, confirmation or rejection of change requests, and verify changes in build	Initial	Update	Update	Complete
5	Monitor, audit, and report configuration status	Initial	Update	Update	Complete

**Table F.5** Change request workflow register (example)

CR No.	Investigate CR/Request Information	Approved/Rejected	Notify CR Owner	Prepare and Schedule Work	Implement the Change	Update Configuration Items



**Table F.6** Quality Assurance Plan

Actions:	Follow RUP iterative, risk-driven, use-case, and business-focused software development process model	Tests to verify and validate that the built system met its requirements	Assess the results against similar process plans found out in industry
----------	--	---	--

**Table F.7** Environment plan

No	Task Name	Inception	Elaboration	Construction	Transition
1	Prepare environment for project by selecting and acquire tools and develop project-specific template	Complete			
2	Prepare environment for an iteration by developing and lunching the development case and verify and set up the tool configuration and installation	Initial	Update	Update	Complete
3	Prepare the iteration by developing guidelines for business-modeling, use-case modeling, user-interface, design, programming, and testing.	Initial	Update	Update	Complete
4	Support environment during an iteration	Initial	Update	Update	Complete