|  |  |  |
| --- | --- | --- |
| **Technique** | **Chapter** | **Description** |
| Identify user classes that have process that might need to be automated | Chapter 6: “Finding the voice of the user” | - A useful technique for this is a collaboration pattern (Ellen Gottesdiener )  - The external entities shown outside your system on a context diagram are candidates for user classes.  - A corporate organization chart can also help you discover potential users and other stakeholders (Beatty and Chen 2012). |
| Create or extend data models for information that is being handle manually | Chapter 13: “Specifying data requirements” | - Used data model is the entity-relationship diagram or ERD. If your ERD represents logical groups of information from the problem domain and their interconnections, you’re using the ERD as a requirements analysis tool.  - An analysis ERD helps you  understand and communicate the data components of the business or the system, without implying  that the product will necessarily even include a database. |
| Create a roles and permissions matrix to capture security requirements that previously were enforced manually. | Chapter 9: “Playing by the rules” |  |
| Identify business rules that must be automated when processes they affect are automated. | Chapter 9, “Playing by the rules” | -“Common knowledge” from the organization, often collected from individuals who have worked with the business for a long time and know the details of how it operates.  -Legacy systems that embed business rules in their requirements and code. This requires reverse-engineering the rationale behind the requirements or code to understand the pertinent rules. This sometimes yields incomplete knowledge about the business rules.  - Business process modeling, which leads the analyst to look for rules that can affect each process step: constraints, triggering events, computational rules, and relevant facts… |
| Create flowcharts, swimlane diagrams, activity diagrams, or use cases to show how users currently perform tasks and how they will perform them after automation. | Chapter 8: “Understanding user requirements” and Chapter 12: “A picture is worth 1024 words” |  |
| Use data flow diagrams (DFDs) to identify processes that could be automated, and create new DFDs to show how newly automated processes interact with existing parts of the system. | Chapter 12, “A picture is worth 1024 words” | - A top-level data flow diagram represents how a business process handles data at a high level of abstraction. Swimlane diagrams show the roles that participate in executing the various steps in a business process flow.  - Refined levels of data flow diagrams or swimlane diagrams can represent business process flows in considerable detail. Similarly, flowcharts and activity diagrams can be used at either high or low levels of abstraction, although most commonly they are used to define the details of a process |
| Adapt business processes to permit use of a COTS solution. | Chapter 22, “Packaged solution projects” | - COTS packages are usually selected because implementing and maintaining them is expected to be less expensive than building custom software. Organizations need to be prepared to adapt their business processes to the package’s workflow capabilities and limitations. This is different from most development projects where the software is designed specifically to accommodate existing or planned processes. |
| Create trace matrices to map process steps to requirements | Chapter 29, “Links in the requirements chain” | Table 29-1 shows how each functional requirement is linked backward to a specific use case and forward to one or more design, code, and test elements. A design element can be something like an architectural component, a table in a relational data model, or an object class. Code references can be class methods, stored procedures, source code file names, or modules within a source file. Including more trace detail takes more work, but it gives you the precise locations of the related software elements. |