Chapter 4. **Roles And Responsibilities**

Who are the people involved in this process?

In this chapter, I introduce the reader to the people that participate in the process.

Everything that is produced is produced for a specific audience. The people who require this information play a role in the release of the product. Each role has their own responsibilities and each activity they are responsible for requires different information. This is why the roles that play a part in the Quality with Agile through Pictures process are defined and the benefit that they gain from the information that they are provided is identified.

Defining roles allows us to justify the activities in the process. An activity that produces information without a beneficiary is an activity whose purpose needs to be questioned.

4.1 **Actors**

An actor describes the characteristics of a group of people (systems, if the function is automated) performing a role. Actors capture the responsibilities and activities performed by the people playing that role. In this book, the actors are the people interacting with the Quality with Agile through Pictures process. These actors may be responsible for the activity, supporting the activity, or merely gaining information from the activity.

In the process diagrams, the line connecting the actor to the activity identifies the role that the actor plays in an activity.

If the line has an arrowhead entering the activity as show in Figure 3, then the actor is responsible for the activity.

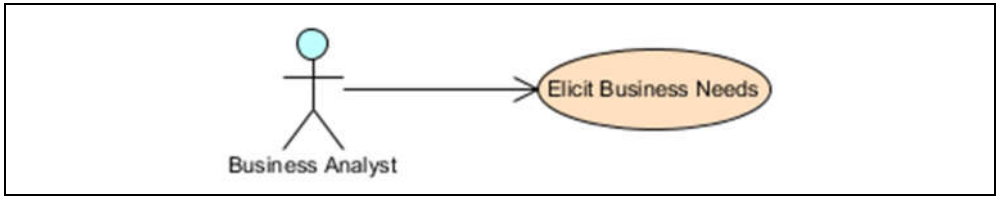

Figure 3 – Primary Actor

The role that is responsible for the activity is known as the primary actor.

If the line connecting the actor to the activity has no arrowhead, as shown in Figure 4, then the actor is supporting or receiving information from, the activity.

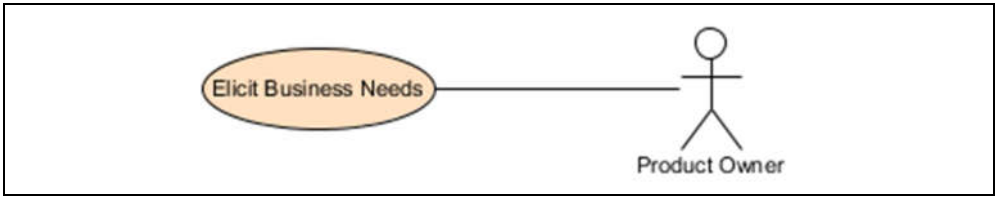

Figure 4 – Secondary Actor

A role that supports or receives information from the activity is known as a secondary actor.

4.2 **The Customer**

The customer is external to the process. The customer represents all stakeholders with knowledge that may be used to derive product requirements. These stakeholders include business sponsors, business subject matter experts, current business process system users and anyone with knowledge of the current business process.

4.3 **Actor Responsibilities**

Everybody involved with development is responsible for quality. Quality starts from day 1, with product planning and vision. Certain quality attributes can be made the responsibility of certain actors. It is assumed that the primary actor is responsible for ensuring the quality of the outputs from an activity. However every role especially the secondary actors, should be contributing to that quality and not be afraid to ask questions concerning quality.

The actors used in this book and their primary and secondary responsibilities, are described in the following sections. The description includes a list of quality attributes that are the responsibility of the actor.

- **The list of quality attributes is not intended to be complete. Feel free to add your own.**

4.3.1 Business Analyst

The business analyst is the go to person for knowledge about the product, as it is and what it will be.

The business analyst is responsible for requirements and may serve as a proxy for the system user or Subject Matter Expert (SME). They will be able to explain the purpose and reason for requirements, and will be able to contact subject matter experts in order to obtain more information.

The business analyst understands the dependencies between system components and is able to assess impacts that are the result of a change to the product.

The business analyst may play the role of a product owner on a Scrum project. The business analyst provides support wherever it is needed. This includes Scrum meetings, testing and deployment. The business analyst gets additional information from the UI design and from the design of the system architecture.

4.3.1.1 Quality Attributes

The business analyst is responsible for accurately specifying the requirements of the product. These requirements may be in the process of specification, they may be in development, in testing or deployed to a customer. The quality of these requirements is determined by the following attributes:

- Abstraction – Is the requirement written using the correct language and level of detail, for its audience?
- Accuracy - How well do I understand the customer needs?
- Analysis – Is the right amount of investigation being invested into this requirement?
- Necessary– Are the requirements design independent and include only useful information?
- Independent – Is the requirement addressing the need of a specific stakeholder role?

♦ **Note that I use the term stakeholder role, and not stakeholder. I do not write requirements for individuals, I write requirements for a specific role.**

- Specification – How rigorously do I need to write the requirement?

The answer to some of these questions varies by the business need that is being satisfied. A well-understood requirement may not need the same level of detail as a brand new feature. More complex requirements with multiple dependencies will need a lot more analysis than a requirement with few dependencies.

4.3.2 Deployment Manager

The deployment manager provides continuous delivery and product release support, as necessary. Their services are on-demand and they strive to meet customer demands within the scope of the product specification.

♦ **Customer support should only assist customers with support that is within the bounds of what the system was built to do. Out of scope support requires approval for additional budget.**

They manage the configuration of software builds and will deploy software releases to the customer.

The business analyst provides support to deployment in the form of product knowledge, as required.

4.3.2.1 Quality Attributes

The deployment manager oversees a team of engineers whose primary responsibility is to keep the customer happy. Quality is measured by the following attributes:

- Customer Support – Is the system performing to the best of its ability in order to satisfy delivered customer needs?
- Release – Is system downtime minimized as a result of changes to the software or hardware?

4.3.3 Development Team

Because most projects that I have been involved with do not have anyone playing the Scrum master role, the Scrum master is included in the development team. This is particularly true when there is only one development team on the project. In this case, the Scrum team polices itself.

The development team creates the software that causes the software to satisfy its user stories. The output from the development team is an incremental software build.

The business analyst supports the development team on a daily basis. The business analyst attends all Scrum ceremonies and addresses development issues as necessary, in order to ensure the success of the current sprint.

4.3.3.1 Quality Attributes

The development team is primarily responsible for delivering software that matches the product requirements. Quality attributes that the development should consider include:

- Accuracy - How well does the software meet its requirements?
- Compliance - How well does the software meet the teams coding and design standards?
- Efficiency – Could the software be made easier to understand, such that it takes less time to make modifications?
- Extensibility - How easy will it be to change the software if there is a change to the requirement?
- Performance - Could the software be written differently, such that the system has better performance while still satisfying its requirements?
- Stability – What is the likelihood that the software will need to be rewritten as a result of new requirements?

4.3.4 Product Owner

The product owner is a stakeholder who verifies the accuracy and priorities of the product's user stories. The product owner ensures the timeliness of

delivery, understands the cost, and benefits to the customer of system functionality. The product owner may play the role of proxy for the project sponsor or other business stakeholders.

The business analyst supports the product owner by analyzing and detailing user stories that capture business needs.

4.3.4.1 Quality Attributes

The product owner's primary responsibilities are to understand the business and their needs and be able to recognize features that give the most benefit to the customer.

- Priority - Do we have our priorities correct?
- Value - Is value being provided to the customer?

The product owner provides support to all activities including attending Scrum meetings.

4.3.5 Quality Assurance

Quality assurance is an independent group of workers who are usually associated with testing. However, in order improve the likelihood of successful testing (hence reducing the number of found defects), quality assurance plays a secondary actor role in most activities. Quality assurance needs to understand the requirements, the architecture and the user interfaces. They should contribute whenever questions about quality arise at any stage in the product life-cycle.

The business analyst assists quality assurance by detailing acceptance criteria and specifying data, which can be used when formulating test cases.

4.3.5.1 Quality Attributes

Quality assurance is responsible for validating the quality of the product prior to release to a customer. Quality attributes they should consider include:

- Accuracy - Can I write accurate test cases from the requirements?
- Documentation – Are the user instructions accurate?
- Performance - Is the platform going to meet performance requirements?
- Regression - Did anything get broken since the last time it was tested?
- Security - Is the product meeting user access needs?
- Validation – How well does the delivery satisfy the functional requirements?

Quality assurance verifies quality through all stages of development. This includes fully understanding the requirements (user stories and acceptance criteria) and ensuring that defects are properly captured and prioritized.

Quality assurance is represented at all Scrum meetings. They provide information about product defects and take away information that can assist with writing test cases.

4.3.6 Solution Architect

The solution architect is a technical expert with knowledge of enterprise architecture. They know how to combine and integrate system components to provide a platform for a solution.

The architect has knowledge of available technology and maintains technology skills in order to identify the best solution from various alternatives.

The solution architect is responsible for ensuring that the platform design provides the best performance and that it is reliable.

The business analyst can assist the solution architect by maintaining a model of the product architecture.

4.3.6.1 Quality Attributes

The solution architect is primarily responsible for the hardware and software that will host the product. Quality attributes they should consider include:

- Scalability – How easy is it to update the architecture in order to support larger solutions?
- Performance – Is the architecture going to respond according to customer needs?
- Reliability – Is the platform going to be available when the customer needs it?

The solution architect supports the development team by providing technical information about integrating the product with the architecture.

The solution architect provides support to deployment as necessary.

The solution architect assists with the requirements by providing technical information that may restrict the way in which a requirement is implemented.

4.3.7 UI Designer

The user interface designer specializes in the usability and quality of a user interface design.

> ♦ **The UI Designer is not developing the user interface. They are applying best practices and company guidelines to user interfaces.**

The business analyst assists the UI designer by detailing user and system interaction steps (and data) in the user stories.

4.3.7.1 Quality Attributes

The user interface designer is primarily responsible for ensuring that the most user-friendly experience is delivered with the product, within the company's user experience guidelines. Quality attributes they should consider include:

- Attractive - Does the user interface appear professional and well laid out?
- Clarity – Is it clear what the user should do as a result of an unexpected event?
- Consistency – Does the user interface comply with company standards?
- Usability – How long is it taking the user to perform their job?

The UI designer supports the development team by providing them with images, usage instructions and component details, of the user interface screens.

4.3.8 Writer

The writer creates user instruction documentation for both customers and internal stakeholders. These documents may include product user manuals, deployment instructions and even development standards.

The business analyst helps the writer to understand what needs to be documented, with their knowledge of system changes and product design.

4.3.8.1 Quality Attributes

The writer is primarily responsible for product user manuals and release notes. Quality attributes they should consider include:

- Accuracy – Are the instructions are a true representation of the current product?
- Consistency – Do manuals follow company branding and standards?
- Maintainability – Are manuals easily updated as new product versions are released?
- Readability – Is the information written in a language and format that is appropriate for the specific reader?

The writer may participate in Sprint Planning, Sprint Review, Sprint Retrospective and Daily standup meetings. The writer should be considered a member of the development who needs to understand any user stories that are taken into a sprint.

4.4 How Do These Roles Work Together

Why do we separate workers into roles? Why can't everyone do anything that is required at the time that it is needed?

The obvious answer is skillset. Some people are trained to be good business analysts and not so proficient at designing a user interfaces. It is more efficient to have people doing what they are best at.

- **I can describe at least 2 instances where I have had to rewrite a software package that was previously written by someone without knowledge of good coding practices.**

Another reason for specific roles is conflict of interest. Each role has its own different objectives. These objectives are not always going to produce the same results. Some examples of roles that have different objectives are:

- The product owner wants to get working features in front of the customer as soon as possible.
- The business analyst wants accurate requirements and an accurate model of the system.
- The development team wants to complete the stories in the sprint.
- Quality assurance wants to prevent defects from reaching the customer.

Is a developer who is required to test their own code, more or less likely to find a defect in that code than a tester who objective is to find defects?

Would a product owner rather spend time getting new features ready for development or spend their time after a sprint accurately capturing what was produced?

Is the business analyst more interested understanding what the product does or what the product is supposed to do?

Roles with different objectives have to find a way to work together to ensure the success of the product, while their own objective may be compromised.

4.5 **Conflicts Of Interest**

Below are just a few examples of roles that need to compromise their objectives in order to work with people who have a different objective.

Why can't the business analyst also play the role of the product owner?

The product owner is concerned with deploying the highest benefit to the customer as soon as possible. A business analyst is concerned with delivering the best quality to the customer. Obviously there needs to be trade-offs between the 2 objectives. Delivering 100% quality takes forever, whereas ensuring the minimum amount of quality to pass testing, delivers the product to the customer in the shortest period of time. Neither is an ideal solution and that is why the 2 roles work together to figure out what is the optimal compromise.

Why can't quality assurance be part of the development team?

There is a conflict of in each role's objectives. The development team is driven by schedule. They want to complete the user stories that they took into a sprint. Quality may be compromised in order to meet their delivery promises. Quality

assurance needs to determine if too much quality was omitted during the sprint. Both parties need to work together to ensure that maximum functionality is delivered during a sprint, while satisfying the expected quality standards for the software.

Why isn't the UX developer part of the development team?

User interface design is based on company-wide standards. Development is focused on the product at hand. UX design needs to take into account not only development needs, but also sales, marketing and other products using the same brand. The user interface design should meet company standards, but developers should be allowed to make recommendations for UI details where it reduces the development effort or improves usability.

4.6 **Development Cycles**

Timing of delivery is another reason why separate roles are needed. Scrum is based a single development cycle, called a sprint. The development team takes inputs and delivers builds within a sprint cycle. However, the product owner, business analyst, solution architect, writer and deployment manage deliverables are not sprint based.

The product owner is working continuously with the customer and their priorities may change on a daily basis.

The business analyst may spend several sprints working on a single user story.

The UI designer needs to have user interface screens prepared for user stories that are entered into a sprint. UI designs also impact the user stories that are detailed by e business analyst.

The solution architect needs to have the system architecture ready for each sprint. Their deliverables are based on customer demands and the availability of new technology.

Deployment occurs anytime after quality assurance has confirmed that a build is ready for release. Generally, a project has a release cycle that encompasses many sprints.

The writer produces user instructions for software that is part of a release. User instructions are not produced for incremental builds, since these are liable to change too frequently.

It is not feasible to partition to work performed by these roles into sprints or user stories. The business analyst is already continuously working on user stories. We do not need to create a user story that details what user stories the business analyst will be working on during a development sprint. Similar arguments can be made for the work performed by the product owner, UI

designer, solution architect, writer and quality assurance. All of these roles are already working on or from, user stories. We do not need more user stories to plan which stories they should be working on.

4.7 **Flow Of Information**

The chain of information flows from the product owner through to the deployment manger, writer and quality assurance. Figure 5 is a graphical representation of this information flow.

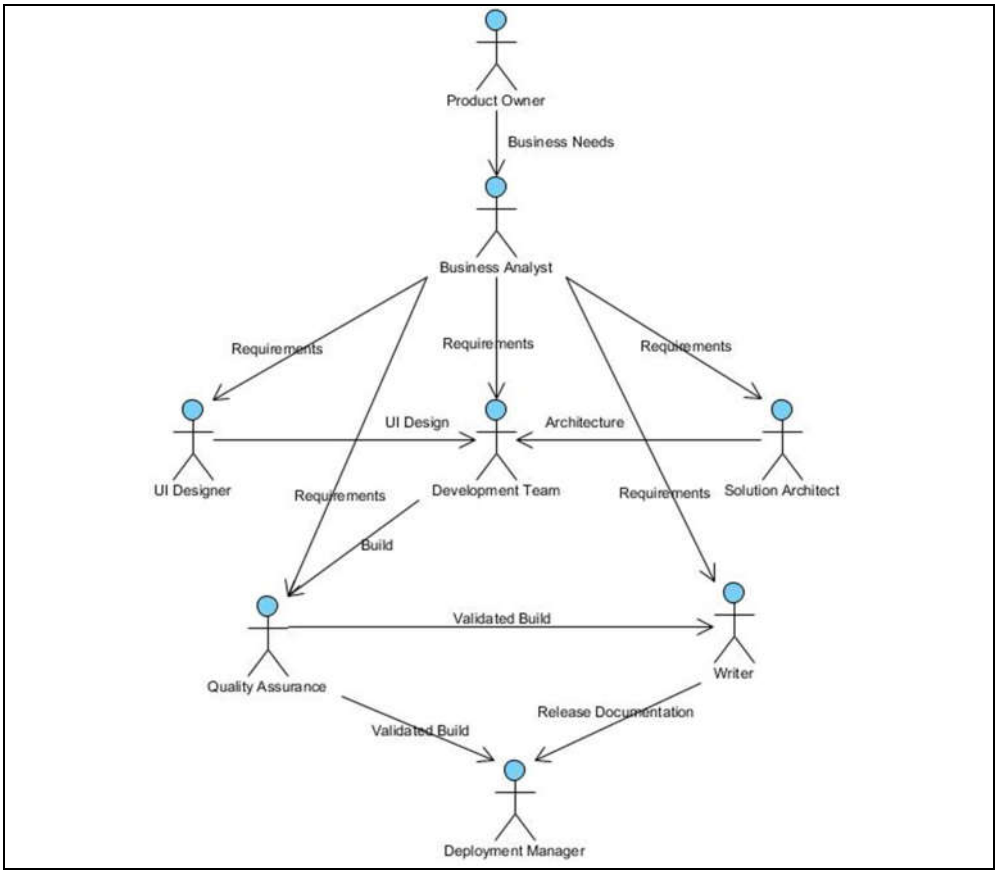

Figure 5 - Flow Of Information

Business needs are prioritized by the product owner. The business analyst converts these to requirements. The requirements are used by the UI designer, solution architect, development team, solution architect, quality assurance and the writer. The UI designer contributes the UI design to the development team. The solution architect contributes to the architecture of the product. The output from the development team is a software build. The software build is tested by quality assurance. The validated software build is documented by the writer.

Validated software and release documentation is deployed by the deployment manager.

Remember that all deliveries of information are incremental. Most information is likely to change after it has been delivered. The only exception is the build. The build is the output from a sprint. A build does not change between sprints. Any changes to the software are delivered in a subsequent sprint via a new incremental build.

- **Ideally, the requirements, UI design and architecture that are input to a sprint will not change either. In reality they do change, which is to be expected. This is fine as long as those changes do not break the sprint.**

This flow of information should not be considered a rule. The diagram shows a generalized flow that rarely follows the relationships exactly as described. In an ideal world, the flow of information would always travel forwards. In reality, when changes are made by a recipient of information, those changes will impact the role providing the information. The flow of information can also go backwards several steps. Sometimes a change made during deployment will even go so far as to impact the business needs, and this propagates through the tree structure again.