



## Notes On What Is A Business Analyst?

RSBA Technology Ltd has provided consulting services to Tier 1 and 2 Investment Banks since 2009. Our focus has been on analyzing the requirements and project managing delivery of complex, multi-site real-time low-latency, high-capacity trading and workflow management systems for eFX businesses within the Bank's Global Markets areas. In the course of this work, and in many years of experience prior to forming the Company, we have gained valuable insights and knowledge on how to effectively design, develop and deploy such systems in a highly competitive landscape.

This Note identifies key tasks and success factors related to the ill-defined project role of a "business analyst". It is based from on my own professional work experience as a business analyst over the past 20 years, across a variety of organisations (software firms, multi-national investment banks) and domain areas within the investment banking industry (buy-side, sell-side, front / middle / back office).

### Tasks

Without attempting at any formal definition, the following list provides a core set of duties that I, and others I have observed in this role, typically undertake in the context of a software project, or work programme. I have tried not to conflate this list with other tasks, such as software development, "classic" project management (think GANTT charts and chairing weekly team meetings) or test engineering (manual test execution and automated test development) which I have often combined in a single engagement.

For the purposes of this Note, I have also excluded higher-level "analysis of the business" tasks (such as reviewing impacts of legislative and regulatory changes), which are usually the responsibility of the business stakeholders (albeit carried out by people called business analysts).

- Analysing current state (current performance levels, audit points etc) and operating model of all or part of the business in scope of the project. Identifying problems, root causes and subsequently helping to define the target state and operating model. In the context of a typical software project at an investment bank, this would involve involvement with and challenging of business stakeholders as a prerequisite for initiating a change project.
- Business requirements elicitation:

- Specific elicitation techniques, for example: user interviews, legacy systems, work shadowing, brainstorming, focus groups, questionnaires.
- stakeholder identification,
- existing system constraints,
- available budgets and resources
- Authoring business requirements documentation of all types (including conducting reviews, ensuring sign-offs, version control etc). There should be (though often is not) a clear separation between the actual business needs, and target system requirements for the given project (scoped & budgeted). Sometimes, the business stakeholders already have a very clear, documented set of business goals and needs. Other times, part of the business analysis role is to understand and document these before embarking on the more detailed system engineering tasks. It should always be a critical business analysis task to question and challenge the business stakeholders understanding of their own needs (or to explicitly document that such definitions are not available and then escalate).
- Creating and documenting possible technical solutions (“Design Ideas”) from stakeholders, team members and from own experience
- Detailed functional and logic design (e.g. Domain Driven Design, UML, abstraction layers, "functional specs").
- UI mock ups, working UI prototypes (e.g. HTML)
- Writing user guides, tutorials
- End user training
- Testing script authoring (but not test execution itself)<sup>1</sup>:
  - Functional test planning and scripting
  - Performance test planning and scripting

---

<sup>1</sup> Note: My colleague, Dick Holland, points out that effective testing requires a different set of specialised skills, which I completely agree with (and possibly the subject of a ‘What is a Tester?’ paper in the future!). However, I felt I had to include this as a core business analyst task in the list because that directly reflects my own experience. I cannot recall a single project I have worked on where I have a major element has involved test authoring, if not test execution.

- UAT test authoring and review with stakeholders
- BDD, TDD using automated testing. Test first principles.

### Skills

Key skills required to be effective:

- Relevant domain knowledge. Whilst a good business analyst should be able to transfer techniques (elicitation, documentation etc) between problem domains, the most effective business analyst on any single project will be someone with those general skills, **and** a subject matter expert (SME) in the domain itself<sup>2</sup>.
- Relevant technical knowledge (you don't have to be a technical expert, but at least need to have working knowledge of generic capabilities of relevant technical architecture, in order to effectively propose design ideas and assist the project team in estimating relative cost)
- Ability to systematically organise information and explore a problem space. Why is this so important? In order to ensure both the full breadth of scope of the problem area has been addressed, and to explore requirements in sufficient detail to ensure all stakeholder needs are met by delivered solutions. In my direct experience, people with scientific or formal engineering backgrounds tend to do this well, because they have been trained in doing this.

The contribution of a business analyst role to overall project delivery, and the timing of those separate tasks, greatly depends on the project methodology employed. The table below summarises how those contributions to major project tasks may vary across project types I usually encounter:

	Waterfall	Agile	Evo
<b>Requirements Elicitation</b>	Up-front requirements & solution analysis then lock off scope. Contribute to change management process	User stories / epics	Identify initial critical stakeholder objectives, functions, constraints, and resources for each cycle.
<b>Requirements Capture</b>	Primary author of the "business specification"	Document use cases, epics	Iteratively update specs in each cycle

<sup>2</sup> Dick Holland writes: "They (BAs) have to know the business in some depth in order to conduct effective elicitation (business users rarely know what their real requirements are, and almost always descend into providing design solutions masquerading as requirements).

Being able to talk the same language as the business stakeholders is a powerful aid to elicitation – not least because business people won't dismiss a BA who clearly understands their business well."

<b>Functional Design</b>	Primary author of the “functional specification”	Document use cases	Contribute design ideas as part of team.
<b>Technical Design</b>	Review traceability to functional and business specifications	Review coverage of use cases	Review traceability
<b>Development</b>	Provide support to developers	Provide support to developers	Provide support to developers
<b>Testing – Component / integration</b>	Write (manual, automated) test scripts up front (the “V-model <sup>3</sup> ”)	Write (manual, automated) test scripts for sprint features, BDD <sup>4</sup>	Write test scripts for Evo Step implemented changes, BDD
<b>Testing – UAT</b>	Author UAT scripts directly, or provide support for SMEs	Contribution to end of sprint user validation process	Contribution to end of Step user validation.

## Conclusion

The purpose of this Note was to clarify the role of a “business analyst” in the context of a typical range of software projects by identifying the typical work tasks involved and associated key skills. In practise, multiple members of a team, none of whom may have the title of “Business Analyst”, may fulfill some of these tasks. Conversely, I have occasionally met those who have the title, but exhibit few of those skills. The most effective practitioners recognise that the core tasks of requirement elicitation and capture are actually rooted in core engineering disciplines and principles<sup>5</sup> of quantification, decomposition (of complex things into simpler, atomic statements) and validation.

## Acknowledgements

Thanks to Dick Holland ([dick@xegetix.com](mailto:dick@xegetix.com)) for commenting on an earlier draft of this paper and providing some valuable insights

<sup>3</sup> See [The Death of the V-Model](#) for a good description why this (& Waterfall itself) is not a good approximation for managing real-world complex engineering projects

<sup>4</sup> Behaviour Driven Development (see [Agile Alliance](#))

<sup>5</sup> See Gilb “Eternal Principles” <http://gilb.com/dl537>