



Scrum & Business Intelligence

Actionable insights from the trenches

Barry Overeem, Sander van Schaik

Business Intelligence (BI) projects can be seen as complex, where the amount of unknown requirements and technologies exceeds the known. BI projects are complex due to fast changing information needs and priorities, existence of many users/customers, availability and quality of data, different systems to extract source data from, and continuously changing technologies. Implementing Scrum with a BI-environment, with a focus on creating actionable insights, is therefore challenging.

The main question we'll answer in this whitepaper is:

How to make sure Scrum and Business Intelligence is a winning combination?

To answer this question, we will share our 8 most important actionable insights with you. Hereby we will focus on individuals, collaboration and the process. Not all insights are strictly relevant for BI; we've just discovered them within a BI context.

Our actionable insights:

1. Stop analyzing, start visualizing
2. Ensure the Product Owner is a leader
3. Invest in test automation
4. Create a cross-functional Business Intelligence team
5. Ensure the BI Architect applies servant leadership
6. Incorporate DevOps practices to get more done
7. Continuously validate the quality of the data
8. Organize a stakeholder-friendly Sprint Review



1. Stop Analyzing, Start Visualizing

The problem with complexity is that it's based on assumptions:

- The customer knows what he wants
- The developers know how to build it
- Nothing will change along the way

The reality is however:

- The customer discovers what he wants
- The developers discover how to build it
- Things change along the way

A pitfall is to spend too much time on analyzing the root cause of the problem or to understand the customer's request. Sure, you need some time to understand the situation. But you will only know if you really understand the customer by building the actual increment. No matter how small this increment might be, it's the only way to **validate your assumptions and discover the impact** by visualizing the results.

The sooner you validate your assumptions the better. The actual customer is the best person to approach for validation. The Product Owner, representing the customer's voice, is also a good option. If the team finishes a Product Backlog Item (PBI) during the Sprint, **ask for validation immediately**. Don't wait until the Sprint Review, because you don't want any surprises during this event.

So in short: stop analyzing, start visualizing, and validate the PBI's as soon as possible.

2. Ensure the Product Owner is a Leader

A Product Owner is crucial for the success of the Development Team. Also within the context of Business Intelligence, the Product Owner is responsible for representing the customers voice, creating a compelling product vision and providing clarity for the Development Team on the items to be worked on. The Product Owner is an entrepreneur for his product, has a keen eye for opportunities, and focuses on business value and the Return on Investment and acts proactive on possible risks and threats. Moreover, **the BI Product Owner should also understand BI-concept and know how to "play" with data.**

Finding a Product Owner that is a 100% match with this description is difficult. Finding someone with in depth BI knowledge isn't the problem, but someone that also has affinity with data and is able to translate a product vision into clear Product Backlog Items is more difficult to find.

Given the described complexity, the main question is:

"How to deal with the scarcity of the ideal Product Owner?"

When you're having difficulties finding the Product Owner that matches all the described responsibilities, make sure the person at least has the mindset and behavior of a leader.

A leader:

- Enables those doing the work to contribute their full talents and capabilities to generate value for customers.
- Trusts in the judgement and wisdom of those in touch with customers as to what work needs to be done.
- Trusts in the talents and capabilities of those doing the work to figure out how to do the work in the right way.

The Product Owner understands **the difference between accountability and responsibility**. He can delegate some responsibilities that might be difficult to fulfill (due to lack of time or capabilities). But the Product Owner understands he will remain accountable for the Product Backlog and the success of the Product.

The Product Owner should create an environment that will support him maximizing the value and ensuring product success. Part of this environment is the Development Team, but this can also include business representatives, stakeholders, users, and sponsors.

The Product Owner as a Leader...

- **Works closely with BI business partners.** This can be users, stakeholders, or sponsors. They are in direct contact with them. They truly understand what they desire and are therefore able to help the Product Owner make the right decisions.
- **Utilizes the Development Team's knowledge.** Having in depth functional or technical knowledge is difficult. Although the Product Owner should have sufficient knowledge to make solid decisions, he should be able to trust the team for the minor details. The BI business/information analyst of the team should cover the functional part, the developers the technical part. As a whole, the Scrum Team should have all the knowledge necessary for developing their products.
- **Encourages the Development Team to help with backlog management.** Managing the Product Backlog can be a time-consuming effort. Although the Product Owner remains accountable for the Product Backlog, backlog management can be a team effort. The team as a whole can support the Product Owner with describing Product Backlog Items, adding acceptance criteria, and providing insights on functional or technical questions.

In our experience the ideal Product Owner should be a leader. A leader that is able to create a compelling product vision and a trust friendly environment. Trust in the judgement and wisdom of those in touch with customers as to what work needs to be done. Trust in the talents and capabilities of those doing the work to figure out how to do the work in the right way. In such an environment the Product Owner can collaborate closely with business partners and the Development Team with product- and backlog management. **Although the Product Owner remains accountable for the product success, delegation of some responsibilities becomes a valid option.**

3. Invest in Test Automation

Having a focus on quality means it should be clear to everyone what the desired quality is. Setting up a Definition of Done hereby is a useful tool. Gunther Verheyen describes the Definition of Done as follows: “The empiricism of Scrum only functions well with transparency. Transparency requires common standards to work against and to inspect upon. **The definition of done sets the standard for releasable¹.**”

The Product Owner should accept the Product Backlog Items during the Sprint. This prevents any undesired surprises during the Sprint Review. This also emphasizes the importance of clear acceptance criteria on which the functional- and technical tests are based. Automating these tests should be part of the Product Backlog Item. This will increase the total amount of time necessary to complete the item, but it will eventually support rapid validation by the stakeholders. **By running tests every night, you’ll ensure a daily quality check.** The most important and difficult part is creating a habit of fixing the failed tests continuously. Because only then you’ll create truly done features.

Besides this, you’ll be able to perform a regression test. **Regression testing will ensure that changes didn’t introduce new errors.** This offers you faster feedback about the quality, a smooth flow through the DTAP environment and most important: creating validated business value every Sprint!

4. Create a Cross-Functional BI Team

The ideal Scrum team consists of a Product Owner that represents the customer’s voice, a Scrum Master with a focus on the process and the team, and a self-organizing, cross-functional Development Team.

We’ve experienced that cross-functionality within a BI-context should contain at least a mixture of team members with the following skills:

- Someone with strong **domain knowledge** able to translate the functional request towards technique. Consider this person as the lubricant between the Product Owner and the other team members. Within BI it’s a common pitfall to spend too much time on analyzing. This person should have a keen eye on finding the right balance.
- Someone with a strong motivation to continuously **improve the (development)process** and strives to minimize redundancy by setting up (test)automation. BI development is a chain of process steps; too much focus on one step is killing for the desired product. Making mistakes is just like every day, normal and useful, but making the same mistakes often means waste of time.
- Someone with **architectural knowledge**. Often the architect isn’t part of the Development Team but someone that is available to support the team whenever necessary. To become a true self-organizing team having solid knowledge about the architecture is important to ensure a smooth progress of the project. Eventually the ‘external’ architect should only be necessary as an advisor. All the architectural knowledge should ideally be available within the team.

¹ <http://www.amazon.com/Scrum-Pocket-Guide-Practice-Publishing/dp/9087537204>

- Someone with **performance knowledge**. Within BI environments we often focus on new functionality before checking the performance. This, while continuously improving the performance ensures faster data processing and thus faster feedback for customers. Therefore optimizing the performance should have main focus from the start. It should be part of the Definition of Done as one of the quality standards the Scrum Team focuses on.

5. Ensure the BI Architect Applies Servant Leadership

Ensuring the BI Architect applies servant leadership is a lesson we've learned within the BI environment, but basically it's relevant for every architectural role. Below we've shared some of the distinctions between the directive and servant architect. These are described in the Dutch article '[The Servant Architect](#)' and we couldn't agree more with them.

The directive architect:

- Is part of the architecture department surrounded by other architects;
- Focuses primarily on the architecture;
- Considers architecture as a product delivered by one or more architects;
- Uses architecture principles and modeling techniques within communication;
- Considers the architect as a guardian that ensures everyone follows their defined approach;
- Creates and guards the architecture.

The servant architect:

- Is part of the Development Team and hereby cooperates and collaborates intensely with them;
- Focuses primarily on the stakeholders;
- Considers architecture as the result of collaboration with the Development Team;
- Uses common sense and simplicity as the most important elements of communication;
- Considers the quality of the architecture as a shared responsibility of the entire Development Team;
- Initiates, facilitates and coordinates the process.

A servant architect can act as catalysis for the Development Team propelling them to unparalleled heights.

The directive architect however who focuses on control will only create fear and uncertainty within the Development Team and hereby act as the millstone on the necks of the Development Team. It may be obvious that we prefer the servant architect.

6. Incorporate DevOps Practices to Get More Done

The goal of a Scrum Team is to deliver a valuable releasable increment every Sprint. Ideally the Development Team has all the capabilities to deliver this increment towards production whenever the Product Owner deems the increment 'production-worthy'.

In large organizations this isn't always possible because Development and Operations are two separate environments. **Often these organizations setup DevOps teams to ensure a smooth application lifecycle management.**

When BI-solutions increase in size, the amount of data also increases and performance becomes key. The feedback loop and hereby the functional acceptance will get slower and slower.

Collaboration with Operations brings some great advantages:

- Operations brings the crucial DBA expertise, including solutions to handle the infrastructure more 'Agile';
- Operations have important skills for monitoring the progress combined with SQL infrastructure knowledge. When the size of the loads increases, this knowledge is necessary to explore the areas where acceleration is possible.
- Faster delivery of features;
- More stable operating environments;
- More time available to add value (rather than fix/maintain).

Therefore, our advice is to incorporate DevOps practices to ensure a smooth application lifecycle management. Consider this as a nice growth path for Scrum Teams.

7. Continuously Validate the Quality of the Data

When starting a new BI project, the team should attempt to deliver a valuable increment from the first Sprint on. This might be a very small feature to see if the plan the team drafted during the first Sprint Planning is sufficient. Although this is very difficult to do during the first Sprint, it encourages a mindset of **releasing early and often**. Together with realizing the first (small) features, the team should focus on gathering knowledge on the availability of high quality data and performance from the start. **BI is all about data;** it's gathered for creating valuable, actionable business insights. Validating the quality of the data and performance should be done continuously. The Development Team should collaborate with the Product Owner to determine what 'good enough' exactly means. Is the data complete, accurate, available and up-to-date? Capturing these quality and performance standards in a Definition of Done hereby is a good practice.

The goal of BI is to create insights. These insights will be truly valuable when they've become actionable and get a follow up by the customers/stakeholders. **Having trust in the quality and reliability of the data hereby is key.** To build a solid foundation of trust the BI department should have comprehensive data quality management to show the status of data quality across all the stages, from source systems and ending with the use of reports.

8. Organize a Stakeholder-Friendly Sprint Review

The Sprint Review is the ideal moment to inspect the delivered increment, and release it to production if the Product Owner finds it useful enough. Nothing is more motivating for the Development Team than attending a Sprint Review with truly engaged stakeholders eager to see and discuss the Sprint results.

A common pitfall however is demonstrating the BI deliverables in such a technical manner the stakeholders are left behind clueless. Of course, some stakeholders might also be interested in the technical composition of the BI products, but often they are more interested in the business value.

A feature-driven approach with a visible and understandable frontend is therefore also recommended within a BI context. This ensures the stakeholders can really inspect the increment and review the delivered business value.

During the Sprint Review with the customers we don't discuss all the technical details.

We encourage organizing a separate session for the Development Teams to share their technical lessons learned and gained insights. This might be an interactive knowledge sharing session with everyone interested in the technical status of the increment. During this session they can have in depth technical discussions about the progress they've made with the BI products and offer demonstrations to the entire BI department.

Besides these periodically company/department wide knowledge sharing sessions, every Development Team will have their own Sprint Retrospective. The goal of this event is to discuss the process, collaboration and engineering practices with the aim to define actionable and committed improvements.