Managing Software Projects by the Buglione-Trudel Matrix

Thomas Fehlmann, Eberhard Kranich
Euro Project Office, Zurich, Switzerland, and Duisburg, Germany
thomas.fehlmann@e-p-o.com
eberhard.kranich@e-p-o.com

Abstract: The *Buglione-Trudel Matrix* (BT Matrix) is a tool based on *Six Sigma Transfer Functions*. It originated from practical experiences in managing software development with agile teams. It maps *User Stories* to *Business Goals*. It manages the development work in a transparent way.

Its primary objectives are visualizing tasks, prioritizing user stories, and assigning them to sprints. Tasks are represented as *Story Cards*, representing a user story, or a portion of it that fits into a single sprint. Story cards carry various software measurements; functional size, testing intensity and applicable test cases, technical debt, effort estimates (by story points), and value for customers. The BT Matrix is used to communicate story cards to the teams, and managers, of the project, and to give feedback to developers how to align their story cards with the overall project goals.

Keywords: Lean Six Sigma for Software, Six Sigma Transfer Functions, Quality Function Deployment, Linear Algebra, Voice of the Customer, Lean & Agile Software Development.

1. Introduction

The BT Matrix helps project teams to organize themselves and adapt easily to changes, both in project goals as well as for technical constraints. It is ideal also for projects targeting the *Internet of Things* (IoT).

The ICT world undergoes a cosmic inflation. Like in the early universe, quantum fluctuations in a microscopic inflationary agile region become the seeds for the growth of structure in the ICT universe, and the universe becomes transparent. This phenomenon, familiar to cosmologists, happens right now to ICT.

The reason is that all "things" become intelligent, receive IP addresses and connect to the Internet. The possibility to create new ICT-based products seems unlimited if there are sponsors to fuel the inflation.

Thus, we'll experience many upcoming projects, connecting databases with real world information, kitchen fridges, cupboards, cars and other transportation devices to meet the needs of people. This tool is intended to enable sponsors, product owners and developers to rapidly agree on what's relevant for users, stakeholders and sponsors. Note that this are agile system integration projects, not limited to software development.

The challenge is to understand the needs of all stakeholders and continuously deliver what they need for keeping the project going, make it profitable and successful. The visual appearance of the BT Matrix is attractive and suitable for communication. However, its organization, and the mathematics behind Six Sigma transfer functions, are not trivial. The reader not familiar with Six Sigma transfer functions should consult Fehlmann (Fehlmann, 2016). An additional problem is that Six Sigma transfer functions work only if a goal has been set, for instance the goal of a software project. This is an issue. Without setting business goals in a sensible way, projects fail for sure, but setting goals for software projects is more difficult than for traditional civil engineering projects. The BT Matrix has been developed by Fehlmann and Kranich since 2011 (Fehlmann & Kranich, 2012).

2. Approach

The idea of writing specification documents and planning software development like civil engineering has now almost completely disappeared. Today, the paradigm for software development is likewise teaching machines what to do. Today's autonomous cars are no longer fully programmed by programmers; they rather learn how to behave in traffic based on neuronal networks (Bojarski, et al., 2016).

Nevertheless, there are still programming teams that write code; yet programming robots that can write code themselves. Obviously, however, such team use an agile approach.

Since tools are abundant, it becomes more and more important what the goal of the undertaking is, not so much how to do it. Thus, setting the goal right is where modern software project management starts. Since this is not easy, software projects fail often.

A method architecture for goal setting combines various tools such as *Analytic Hierarchy Process* (AHP), Kano, *Net Promoter®* Surveys (NPS), *Voice of the Customer* (VoC) techniques and competition analysis with the *New Lanchester* theory. The combined goal profile is the used to prioritize *User Stories* and generate relevant *Test Stories*. All this belong to the field of *Quality Function Deployment* (QFD), see the new series of ISO standards on the *Statistical and Related Methods to New Technology and Product Development Process* (ISO 16355-1:2015, 2015). For applications and best practices, see Fehlmann (Fehlmann, 2016), and other papers of the authors, e.g., Fehlmann and Kranich (Fehlmann & Kranich, 2012). This paper addresses managing a software project, assuming goals have been set.

3. The Buglione-Trudel Matrix

The *Buglione-Trudel Matrix* (BT Matrix) visualizes functional and non-functional aspects of software development, as explained in (Fehlmann, 2016, p. 189) and (Buglione & Trudel, 2010). This combination allows tracking effort spent on non-functional or quality tasks, and compare it to work performed on functionality. In agile development, many functional tasks become apparent during requirement elicitation only; initially, they are hidden behind non-functional, or unspoken, requirements, to become apparent during the sprints only.

3.1 Story Cards

In the BT Matrix, *Story Cards* are placed in a matrix matching *User Stories* to *Business Goals*. In our agile practice, story cards are used in agile development to break down user stories into work items that fit into a sprint. Wording might be different in other specific environments and in other agile teams. User stories are a more common term in agile, whereas the term “business goal” needs some explanation. Denney (Denney, 2005) used the term *Business Drivers* for the goals of product development. Business goals are identified by the *Voice of the Customer* (VoC) method, part of *Quality Function Deployment* (QFD); terms used in *Six Sigma for Software*. They represent the goal of the software development project (Fehlmann, 2016, p. 68) and (ISO 16355-1:2015, 2015).

3.2 The Buglione-Trudel Matrix Visualization

The lower part of the BT Matrix, called the *Cellar*, contains the functional story cards (in white) that do not have specific impact on business goals. The story cards that have specific impact reside in the upper half, the *Sundeck*. The sundeck contains both functional and non-functional, quality-related story cards.

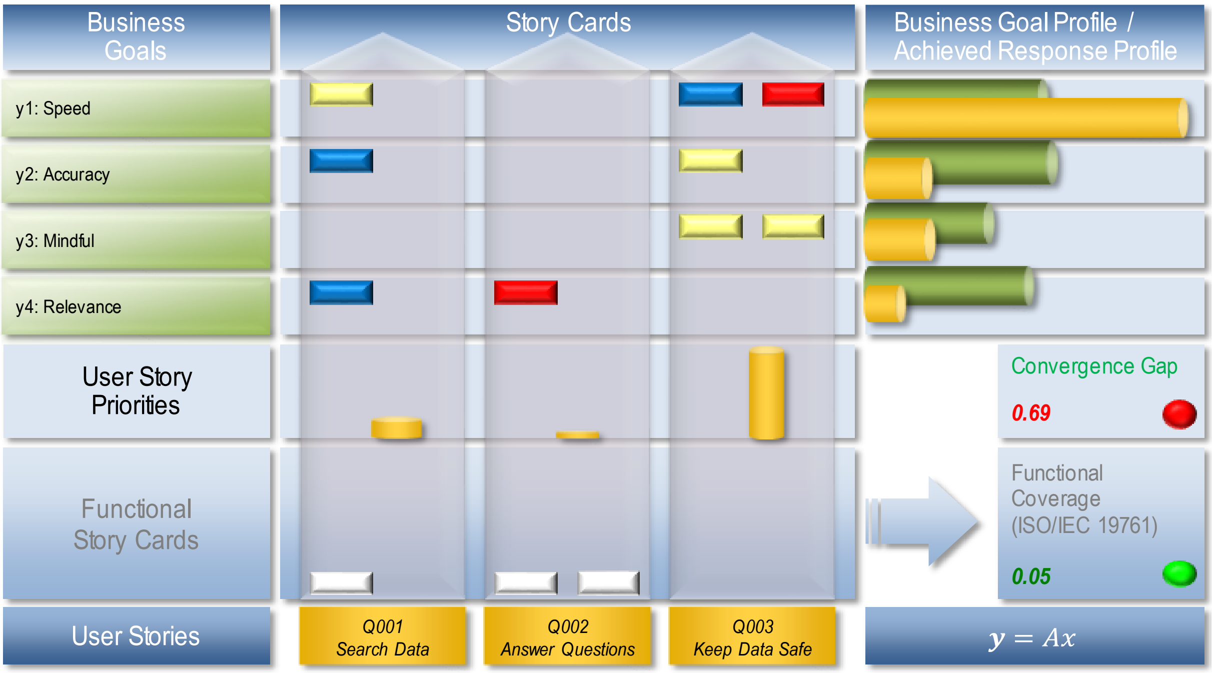


Figure 1: Initial Buglione-Trudel Matrix after the first Sprint; yet missing the business goal profile

Note that story cards can show up in a BT Matrix more than once, since they can specifically impact more than one business goal.

After a few sprints, the development team has adjusted the BT Matrix to meet the goal profile of the business goals (Figure 2) by adding the story cards needed for narrowing the *Convergence Gap*. The convergence gap reflects the Euclidean distance between the goal profile and the profile achieved by the developers:

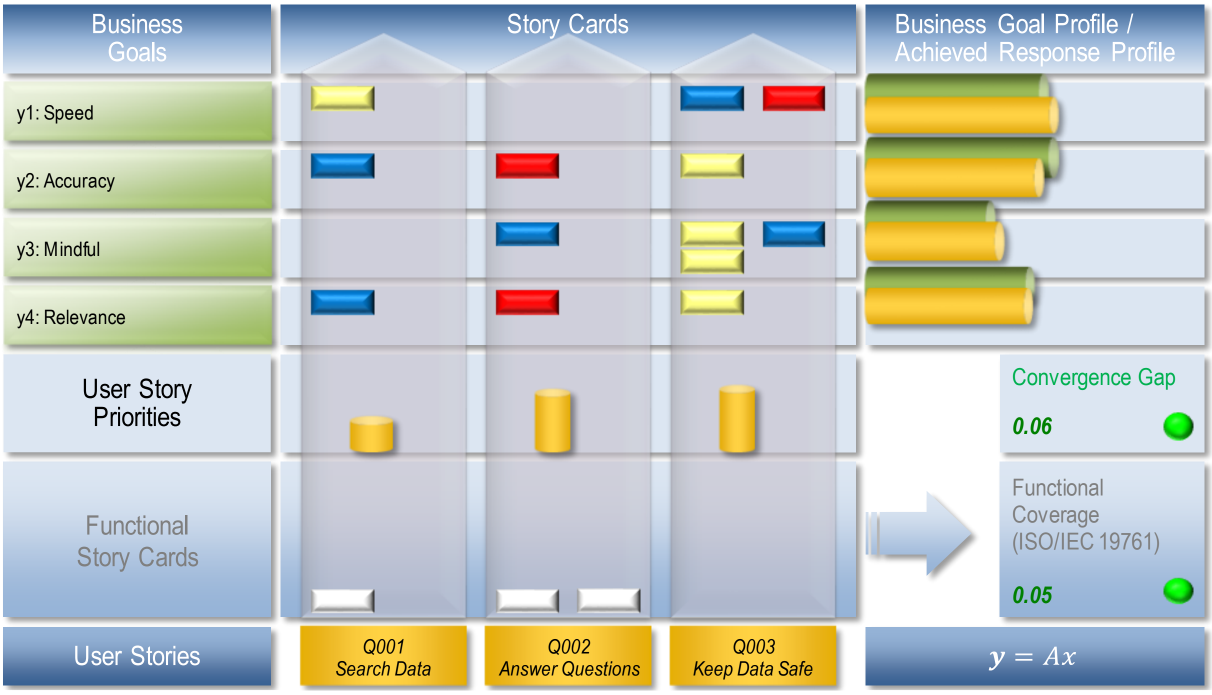
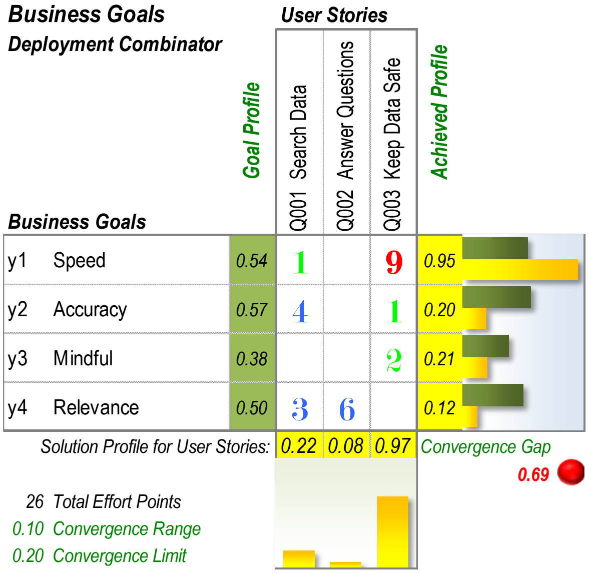


Figure 2: Final Buglione-Trudel Matrix; finely targeting the business goal profile

The BT Matrix contains two transfer functions: one mapping user stories into business goals by means of the business impact recorded on story cards, and another one mapping user stories into business goals by means of the functional effectiveness transfer function. The sundeck transfer function is explicit in the BT Matrix; the functional effectiveness transfer function is referenced in the matrix cellar.

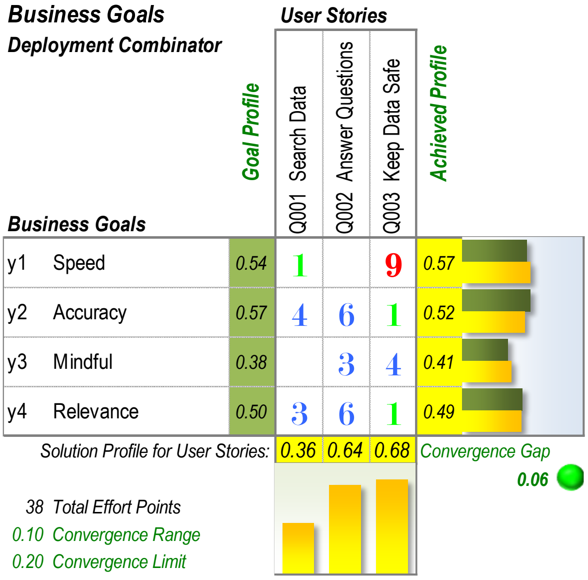
Figure 1 (initial) and Figure 2 (final) show the convergence gaps calculated by the transfer function.



Business Goals
Deployment Combinator

Business Goals		Goal Profile	Q001 Search Data	Q002 Answer Questions	Q003 Keep Data Safe	Achieved Profile
y1	Speed	0.54	1		9	0.95
y2	Accuracy	0.57	4		1	0.20
y3	Mindful	0.38			2	0.21
y4	Relevance	0.50	3	6		0.12
	Solution Profile for User Stories:		0.22	0.08	0.97	Convergence Gap 0.69 ●

26 Total Effort Points
0.10 Convergence Range
0.20 Convergence Limit



Business Goals
Deployment Combinator

Business Goals		Goal Profile	Q001 Search Data	Q002 Answer Questions	Q003 Keep Data Safe	Achieved Profile
y1	Speed	0.54	1		9	0.57
y2	Accuracy	0.57	4	6	1	0.52
y3	Mindful	0.38		3	4	0.41
y4	Relevance	0.50	3	6	1	0.49
	Solution Profile for User Stories:		0.36	0.64	0.68	Convergence Gap 0.06 ●

38 Total Effort Points
0.10 Convergence Range
0.20 Convergence Limit

Figure 3: Initial Sundeck Convergence Gap

Figure 4: Final Sundeck Convergence Gap

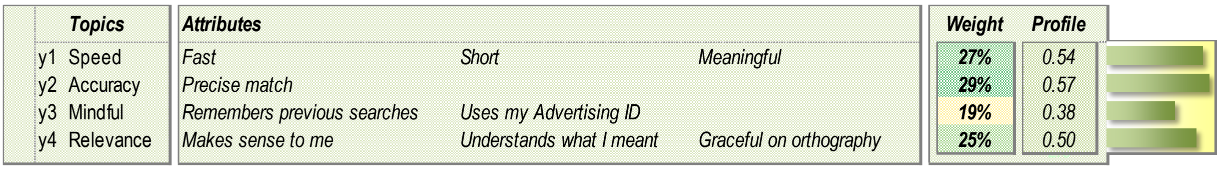
Like in Figure 10, we use Six Sigma transfer functions to calculate whether the development effort achieves the expected goal, namely the business goal's goal profile. The two profiles are regular, see Fehlmann (Fehlmann, 2016, p. 36); that means, if the yellow bar overtakes the green bar, it stands for a relative overachievement. Six Sigma transfer functions define goals by a *Goal Profile*, a normalized vector of length 1 in the Euclidean norm allowing to specify the direction to go in the space of business requirements. Values in profiles are not absolute. In Figure 3, the business goal *y1:Speed* is not necessarily an overachievement; it looks strong because the other topics are much too weak.

The total impact in Figure 1 does not satisfy all business goals and thus does not provide the correct business value. The team can identify weak spots in the BT Matrix, and thus gets an indication where improvements add impact. The goal profile for the business goals shows over- and underachievement by comparing the two profiles.

Completing the final BT Matrix can be done at any time during software product development. Sometimes, business goals change during development.

4. A Sample Project

Our sample project shall search for data, like any Google search; however, we want to provide an extra layer of excellence in user experience, to promote our business. This product quality characteristics is our *Business Goal*.



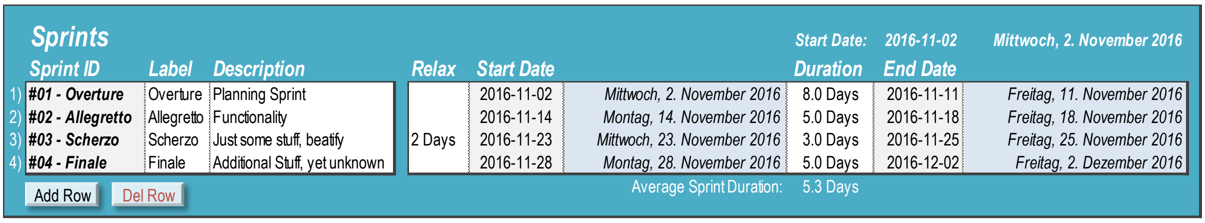
Topics	Attributes			Weight	Profile
y1 Speed	*Fast*	*Short*	*Meaningful*	***27%***	*0.54*
y2 Accuracy	*Precise match*			***29%***	*0.57*
y3 Mindful	*Remembers previous searches*	*Uses my Advertising ID*		***19%***	*0.38*
y4 Relevance	*Makes sense to me*	*Understands what I meant*	*Graceful on orthography*	***25%***	*0.50*

Figure 5: Defining our Business Goals as a Goal Profile

Although the profile is relatively flat, it is enough to assign the story cards in a way that satisfies business goals.

4.1 Preliminary Steps

A holiday calendar takes care of local holidays, allows planning sprints effectively. Compulsory holidays do not add to the duration of sprints. The calendar is user-defined. When defining the team, each member must have a unique nickname. Sponsors and other stakeholders not involved into development do not need such unique nicknames; they are listed here for convenience only.



Sprints

Start Date: 2016-11-02 Mittwoch, 2. November 2016

	Sprint ID	Label	Description	Relax	Start Date		Duration	End Date	
1)	***#01 - Overture***	Overture	Planning Sprint		2016-11-02	*Mittwoch, 2. November 2016*	8.0 Days	2016-11-11	*Freitag, 11. November 2016*
2)	***#02 - Allegretto***	Allegretto	Functionality		2016-11-14	*Montag, 14. November 2016*	5.0 Days	2016-11-18	*Freitag, 18. November 2016*
3)	***#03 - Scherzo***	Scherzo	Just some stuff, beatify	2 Days	2016-11-23	*Mittwoch, 23. November 2016*	3.0 Days	2016-11-25	*Freitag, 25. November 2016*
4)	***#04 - Finale***	Finale	Additional Stuff, yet unknown		2016-11-28	*Montag, 28. November 2016*	5.0 Days	2016-12-02	*Freitag, 2. Dezember 2016*

Average Sprint Duration: 5.3 Days

Figure 6: Sample Sprint Planning

The sprint schedule is defined by

- Its start date;
- The duration of each sprint;
- Additional relaxation days before starting a new sprint.

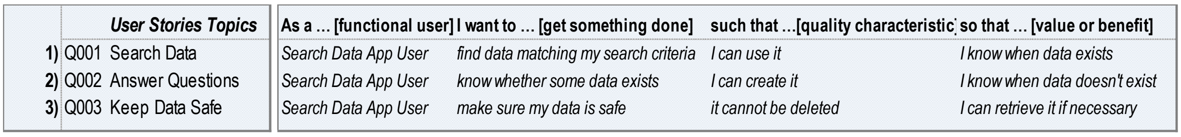
Each sprint can have a label for easier identification. We usually use terms from classical music to set the stage for the sprints (sure, they might change, and most often you must add slow motions, an Andante or some Largo pesante) but such choice depends form the team and the topic involved.

4.2 Get the User Stories

Non-functional requirements most often become part of the quality characteristics part – it is usually not needed to write extra non-functional user stories. Every quality characteristic should refer to some functionality, else it is difficult to verify its existence.

User stories characterize the way how to get some value or benefit, and what conditions must be met. It is usually not needed to write extra non-functional user stories. Every quality characteristic should refer to some functionality, else it is difficult to verify its existence.

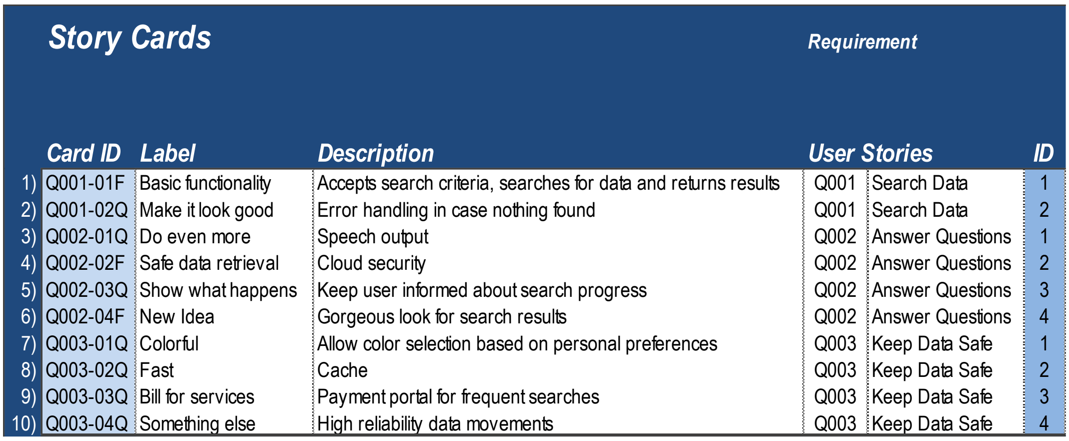
Thus, following Fagg and Rule (Fagg & Rule, 2010), and in extension to what seems to be the standard (Buglione, et al., 2011), we always want quality characteristics to be identified as part of an otherwise functional user story. When implementing functionality, the quality characteristics may give rise to some extra work.



	User Stories Topics	As a ... [functional user]	I want to ... [get something done]	such that ...[quality characteristic	so that ... [value or benefit]
1)	Q001 Search Data	Search Data App User	find data matching my search criteria	I can use it	I know when data exists
2)	Q002 Answer Questions	Search Data App User	know whether some data exists	I can create it	I know when data doesn't exist
3)	Q003 Keep Data Safe	Search Data App User	make sure my data is safe	it cannot be deleted	I can retrieve it if necessary

Figure 7: User Stories

The user stories in Figure 7 are not specific enough to start working on them, thus they translate into the following table of story cards



Story Cards

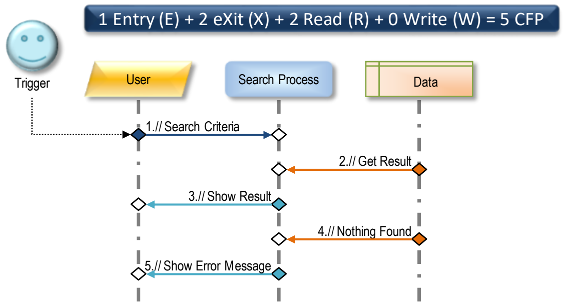
	Card ID	Label	Description	Requirement User Stories		ID
1)	Q001-01F	Basic functionality	Accepts search criteria, searches for data and returns results	Q001	Search Data	1
2)	Q001-02Q	Make it look good	Error handling in case nothing found	Q001	Search Data	2
3)	Q002-01Q	Do even more	Speech output	Q002	Answer Questions	1
4)	Q002-02F	Safe data retrieval	Cloud security	Q002	Answer Questions	2
5)	Q002-03Q	Show what happens	Keep user informed about search progress	Q002	Answer Questions	3
6)	Q002-04F	New Idea	Gorgeous look for search results	Q002	Answer Questions	4
7)	Q003-01Q	Colorful	Allow color selection based on personal preferences	Q003	Keep Data Safe	1
8)	Q003-02Q	Fast	Cache	Q003	Keep Data Safe	2
9)	Q003-03Q	Bill for services	Payment portal for frequent searches	Q003	Keep Data Safe	3
10)	Q003-04Q	Something else	High reliability data movements	Q003	Keep Data Safe	4

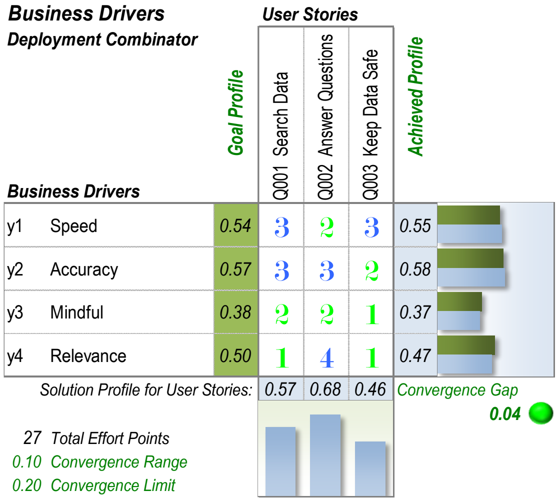
Figure 8: Story Card Table, based on three User Stories

Figure 8 contains sufficient backlog for a small team, to be executed in three to four short sprints. This list corresponds to the final BT Matrix in Figure 2.

5. Functional Effectiveness

Functional effectiveness counts cell by cell in the transfer function mapping user stories to business goals. Figure 9 shows the data movement map that implements the functionality needed for our data search project; Figure 10 maps the data movements onto user stories that depend on them to demonstrate that the functionality meets the goal profile of the business goals in Figure 5. Data movement maps look alike *UML Sequence Diagrams*, except that they do not necessarily show looping and conditional behaviour.





Business Drivers
Deployment Combinator

Business Drivers		Goal Profile	Q001 Search Data	Q002 Answer Questions	Q003 Keep Data Safe	Achieved Profile
y1	Speed	0.54	3	2	3	0.55
y2	Accuracy	0.57	3	3	2	0.58
y3	Mindful	0.38	2	2	1	0.37
y4	Relevance	0.50	1	4	1	0.47
Solution Profile for User Stories:			0.57	0.68	0.46	Convergence Gap

Convergence Gap 0.04 ●

27 Total Effort Points
0.10 Convergence Range
0.20 Convergence Limit

Figure 9: Data Movement Map for the Search Project **Figure 10:** Functional Coverage

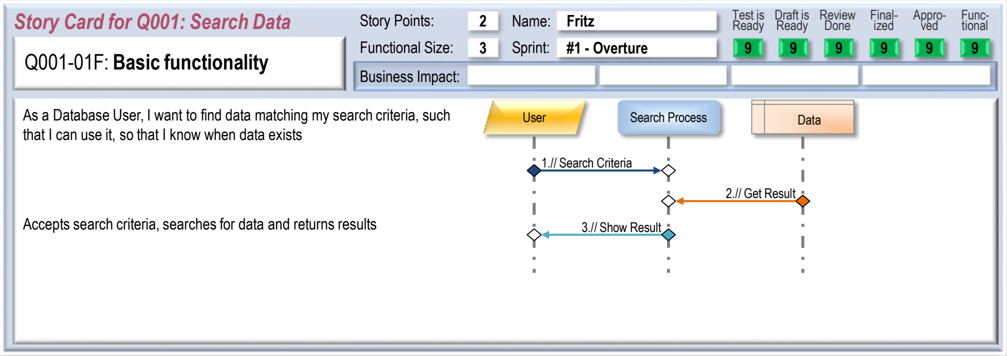
Six Sigma transfer functions, as shown in Figure 10, verify that a given set of controls, here the user stories, are sufficient to produce the response as expected. In the sample project, it means that the software meets the business goals of the sponsor. Thus, if the goal profile for the business goals, and the profile achieved with the functionality implemented by the data movements of Figure 9, are similar enough, the users view is reflected completely and correctly, and no 無駄 (*Muda*, english: *Waste*) is produced. Such software development is not only agile, but *Lean* as well. This is what *Lean Six Sigma for Software Development* exactly means, if due software measurement principles are applied and research is not only based on surveys, beliefs, feelings, and hype. For these topics, consult Fehlmann (Fehlmann, 2016, p. 17).

5.1 Visualization of Muda for Agile Teams

User stories translate into *Story Cards*. A story card fits into one sprint and has only one responsible developer. Ownership of a story card is always with the developer who chose it. In most user stories, several aspects are mentioned that should be separated among developers Typically, there are two different kind of story cards:

1. Story cards that predominantly implement functionality
2. Story cards that predominantly implement non-functional qualities

A story card is a *Functional Story Card* if it has no specific impact on one or several of the business goals; see Figure 11 for an example. They go into the cellar of the BT Matrix. In contrary, story cards that have specific impact are *Impact Story Cards*. These go on the sundeck. For an example, see Figure 12. Impact story cards may well contain functionality; however, most often it is unknown at the beginning of the project, which functionality will be affected. Typically, the development team devises together with the sponsor and product owner what additional functionality to add to impact cards. Thus, total functional size will grow during the development cycle. Other story cards might address dedicated effort not involving any software functionality but required by business.



Story Card for Q001: Search Data

Q001-01F: **Basic functionality**

Story Points: 2 Name: Fritz

Functional Size: 3 Sprint: #1 - Overture

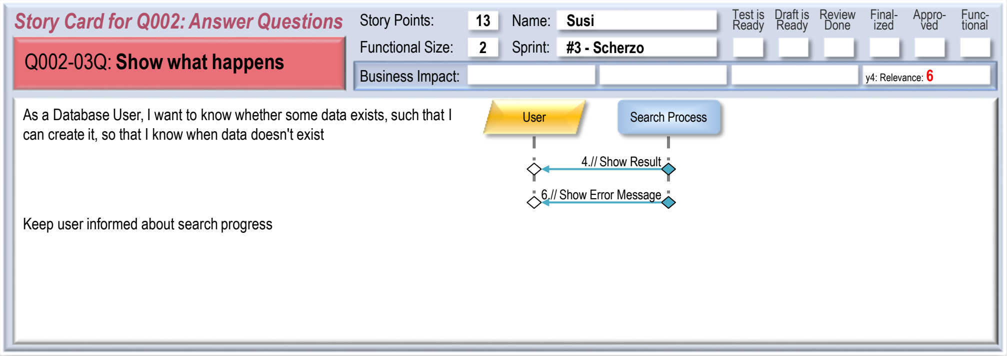
Business Impact:

Test is Ready	Draft is Ready	Review Done	Final-ized	Appro-ved	Func-tional
9	9	9	9	9	9

As a Database User, I want to find data matching my search criteria, such that I can use it, so that I know when data exists

Accepts search criteria, searches for data and returns results

Figure 11: Basic Functionality Story Card; already completed



Story Card for Q002: Answer Questions

Q002-03Q: **Show what happens**

Story Points: 13 Name: Susi

Functional Size: 2 Sprint: #3 - Scherzo

Business Impact: y4: Relevance: 6

Test is Ready	Draft is Ready	Review Done	Final-ized	Appro-ved	Func-tional

As a Database User, I want to know whether some data exists, such that I can create it, so that I know when data doesn't exist

Keep user informed about search progress

Figure 12: Story Card focusing on product quality aspects, with strong impact on business goal *y4: Relevance.*

On the back of the story cards, the unit tests are recorded that that all must be passed successfully before the story card is marked as completed or done, see section *5.2: Six Steps to Completion*.

Story cards record a task description as short as possible, story points assigned by the agile team, functional size counted with ISO/IEC 19761 COSMIC, and specific business impact on the business goals. The team defines specific business impact as the contribution of some story card to one of the business goals. Impact might be zero, if the task is purely functional – a must-be task required by some user story, as in Figure 11 – or if there is no contribution that is specific to one of the business goals identified.

For all impact story cards, business impact is weighted, for instance with 1 to 6 points, to distinguish between low and high impact. The team can agree on a suitable scale. Typical impact values are one (yellow cards in Figure 1, three (blue cards) and six (red cards).

Functionality alone will not have enough impact on the business goals of our sponsor's company, so additional impact story cards will be needed to make the project fly. The sample card in Figure 12 adds no additional functionality yet; however, when it comes to its sprint and implementation, the team might as well invent additional functionality that was impossible to anticipate at project inception stage. Thus, the two data movements appear on the card as suspect to modification or change. Therefore, non-functional requirements play an eminent role for project effort estimation.

5.2 Six Steps to Completion

The *Definition of Done* in Six Sigma for Software follows the *Six Steps to Completion* model (Fehlmann, 2006). It is hardly different from traditional settings in Agile except the strict adherence to *Test-Driven Development* (Fehlmann & Kranich, 2013). Test stories come always first, no story card is approved for implementation without defined tests.

- Test is Ready stage (10%): do we have a unit test to start with?
- Draft is Ready stage (30%): does a full draft exist? i.e., is the deliverable complete?
- Review Done stage (15%): has the deliverable been peer reviewed?
- Finalized stage (20%): usually, some improvement opportunities need a fix.
- Approved stage (15%): team and sponsor agree that the result is complete.
- Ready for Use stage (10%): other stakeholders or users rely on the work delivered.

The Six Steps to Completion measurement method allows for an unbiased assessment of the progress done in a sprint, and early identification of obstacles. It is a proven way to structure daily scrums, and it creates the awareness that completing a draft code is not more 30% of the total work required.

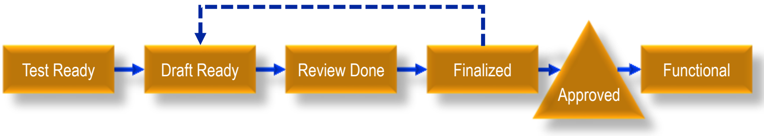


Figure 13: Principle of Six Steps to Completion

5.3 The Kanban Chart

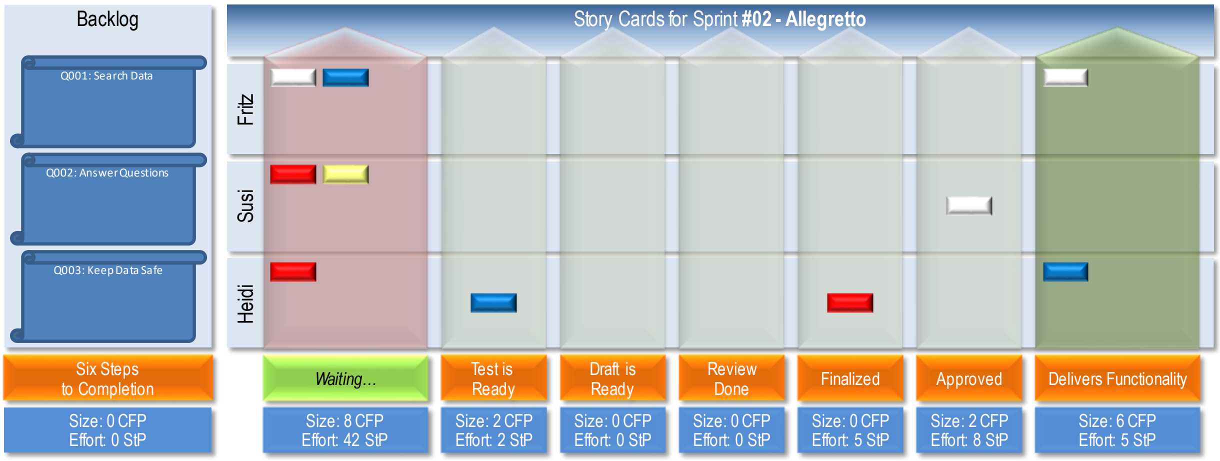

Backlog
Q001: Search Data
Q002: Answer Questions
Q003: Keep Data Safe
Story Cards for Sprint #02 - Allegretto
Fritz
Susi
Heidi
Six Steps to Completion
Size: 0 CFP Effort: 0 StP
Waiting…
Size: 8 CFP Effort: 42 StP
Test is Ready
Size: 2 CFP Effort: 2 StP
Draft is Ready
Size: 0 CFP Effort: 0 StP
Review Done
Size: 0 CFP Effort: 0 StP
Finalized
Size: 0 CFP Effort: 5 StP
Approved
Size: 2 CFP Effort: 8 StP
Delivers Functionality
Size: 6 CFP Effort: 5 StP

Figure 14: Kanban Chart with the Definition of Done for the Second Sprint, showing progress by Story Cards

A Kanban Chart organises sprints and provides a visual overview over the completion rate of a project. Usually, it is on paper and looks somewhat like Figure 14. The story cards are represented by the colours they will get in the BT Matrix explained in section 3, indicating their impact on business goals. The *Backlog* contains the unassigned story cards, or user stories.

The numbers on the bottom column indicate software metrics and story points, estimating effort. Metrics are calculated automatically from the data movement maps in the story cards. The Kanban adapts to number of sprints, team size, number of user stories, and displays all story cards from Figure 8.

The BT matrices in Figure 1 and Figure 2 do not show the details of the cellar matrix in Figure 10, calculating functional effectiveness – for simplicity, but also because developers do not need looking at it, normally. The cellar matrix is significantly simpler, as there is no business impact assessed by experts. It is sufficient to assign data movements in a data movement map to one of the user stories. Following ISO/IEC 19761 COSMIC, this is a side effect of the count.

6. Building the Buglione-Trudel Matrix

The BT Matrix provides a different view on story cards. In contrary to the Kanban Chart, some story cards can appear in more than one cell, in case they support more than just one of the business goals. The teams thus will use the BT Matrix for evaluating priorities rather than just planning the next sprint. The Six Sigma transfer function delivers priorities with respect to the business goals.

Nevertheless, success with the BT Matrix depends on whether the teams are willing and able to do this piece of extra work, assigning business impact to each story card.

6.1 Determining Business Impact for Story Cards

The process of assigning business impact on story cards links to assigning story points. In most cases, story points and impact go in parallel; the more story points the higher is business impact. However, we expect the relationship not linear, since doubling the effort rather seldom also doubles business impact. The BT Matrix is a means to show the team how far away they are from meeting business goals, or customer's needs, when creating ideas how to solve problems encountered.

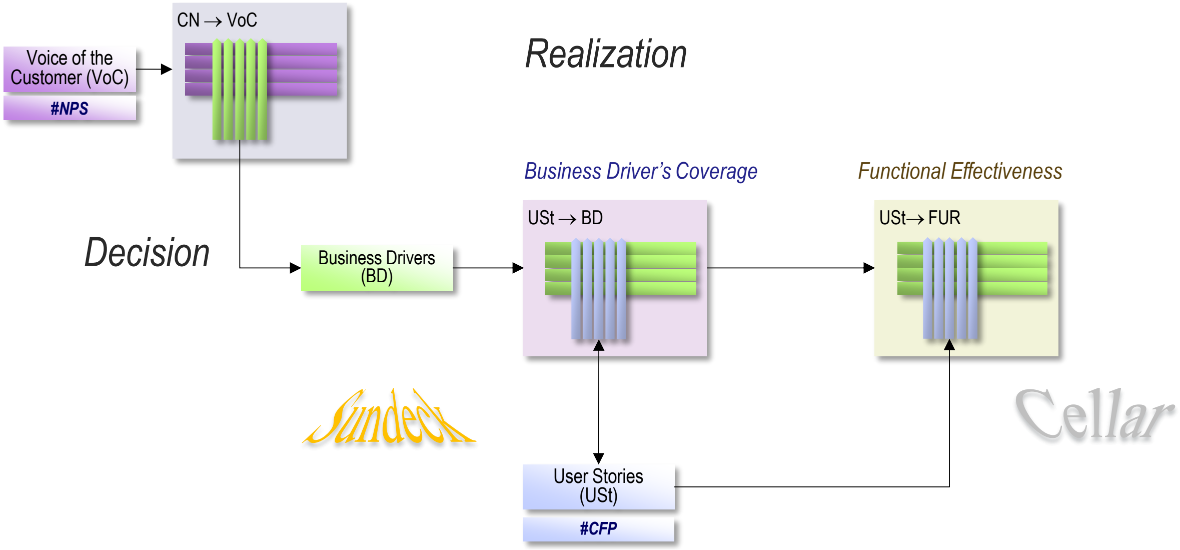


Figure 15: Deming Chain for Agile Software Development

Figure 15 shows the *Deming Chain* for business goals and functional effectiveness. Deming chains is the name used in Fehlmann (Fehlmann, 2016, p. 100) for the value chain between processes that are managed by Six Sigma transfer functions. The two transfer functions can be the same, but usually are not. If they are the same, they convey a system whose business goals are purely functional. Such systems exist but are not the normal case.

The solution profile for business goal's coverage and for functional effectiveness usually are not the same. This is because the two transfer functions represent different measurements. The first solution profile is the goal profile for prioritizing story cards; the second is the goal profile for testing.

6.2 Harvesting Developer’s Intelligence

The development team sees its story card arranged with business goals on the BT Matrix. Initially, response will not match expected response, because one or several of the business goals are not well covered. The convergence gap opens. However, in the practice of the author (Fehlmann, 2016, p. 189), the profile comparison on the right-hand side of the BT Matrix tells easily where the deficit – or the overachievement – occurs.

The matrix cells evaluate as follows: the developers distribute impact points to all story cards recorded in the matrix cells, indicating how much they contribute to each business goal. This happens initially, at planning poker time, and the team might revise it whenever needed. The developers use colour marker buttons that they can distribute; red cards represent value 6, blue cards represent value 3, and light green cards represent value 1. Intermediary values are permitted. Story cards in the “functional” cellar receive no impact points, as functional user requirements are not specific to any of the business goals. They remain white. The team places story cards with specific impact always in the sundeck, although their functionality is part of functional effectiveness.

This visualization is immediately understandable for developers and customers alike. Areas with strong correlation in the matrix contain much red, indicating important focus topics for reaching the stated goal. Areas that contribute less remain with lighter colour or even white. Developers will come up with ideas how to balance the matrix and reduce the convergence gap.

The development team adds story cards according importance for the customer, such that the BT Matrix becomes balanced. The convergence gap metric is a quality indicator for the quality of the software product. A small convergence gap indicates that all business goals are addressed, and none over-achieved.

6.3 Controlling Agile Development

Meeting the customer’s business goals means to implement a solution that fits the goal profile for the business goals. The sundeck transfer function maps the business impact of user stories onto business goals. Business impact are the reasons why non-functional story cards have been included to the user story during one of the sprints. The team chooses and agrees these detailed reasons with the customer, as part of requirements elicitation. They were unknown before in full detail.

The convergence gap measures how well the chosen story cards match the business goal’s goal profile. If the gap opens, the team can identify which aspects need more attention, and place additional story cards. In contrary, if some aspects are over-fulfilled, planned story cards can be removed or new, brilliant ideas rejected just on the fact that they apparently do not add new value for the customer. The sundeck of the BT Matrix serves for balancing the efforts with the needs of the customer, in a well-understood, visual manner. This makes agile software development lean. It blocks waste effectively from becoming part of a sprint.

Obviously, customer values can change, and so can their perception of business goals. This poses no problems as the BT-matric can immediately react to a new goal profile and identify the newly missing parts, and probably also identify overachievements regarding the new goals.

6.4 Controlling Functional Coverage

Meeting functional user requirements is different. Data movements implement functionality, not business impact. Data movements maps describe the functionality needed. The transfer function is simply measurable by counting the number of ISO/IEC 19761 COSMIC data movements that contribute to some user story. The total number of data movement counts in the cells of the cellar matrix is larger than the total functional size, as there are many data movements serving more than one user story. Wasted functionality opens the convergence gap widely. If so, there is the possibility to save effort by removing part of the functionality as originally envisioned. Usually the cellar is more predictable and stable than the sundeck and less prone to adding or changing requirements.

7. Conclusion

The BT Matrix offers a simple and easy tool for communicating goals and achievements to both developers and sponsors. The developers remain in charge and are responsible for success; however, visualization supports communication and decision making. Agile development becomes more predictable because the sundeck of a BT Matrix can be predicted at very early stages of the project by the QFD method. This yields early cost estimates for agile projects, as proposed by the authors, e.g. (Fehlmann, 2016, p. 189).

The approach requires tool support. Agile software development traditionally is paper-based; the team uses paper tags to record user stories and produce story cards. Suitable tools, especially those integrated with code quality and code management tools used in agile, are not yet available. The authors use a prototype software with limited applicability for real software development, outside universities. Nonetheless, it was developed and used for the authors' own experience with agile software development, and there it proved very successful. Developers like to the visualization approach, because it makes it clear for them to explain why they need to add, or throw away, some piece of code. This is how developers can influence business impact.

Future research could focus on converting the BT Matrix software measurements into indicators characterizing software for consumers, for safety of autonomous cars or robots, or for security when adding new appliances in IoT. Quality indicators for consumers are critical for digitalization.

8. References

Bojarski, M. et al., 2016. *NVIDIA Accelerated Computing.* [Online]
Available at: https://developer.nvidia.com/accelerated-computing-toolkit
[Accessed 11 April 2017].

Buglione, L. et al., 2011. *Guideline for the Use of COSMIC FSM to manage Agile projects,* Montréal, Canada: The COSMIC Consortium.

Buglione, L. & Trudel, S., 2010. *Guideline for Sizing Agile Projects with COSMIC.* Stuttgart, Germany, DASMA e.V.

Denney, R., 2005. *Succeeding with Use Cases – Working Smart to Deliver Quality.* Booch–Jacobson–Rumbaugh – Series ed. New York, NY: Addison-Wesley.

Fagg, P. & Rule, G., 2010. *Sizing User Stories with the COSMIC FSM method.* [Online]
Available at: http://sms.amemory.co.uk/wp-content/uploads/20100408-COSMICstories-article-v0c1.pdf
[Accessed 27 February 2015].

Fehlmann, T. M., 2006. *Statistical Process Control for Software Development – Six Sigma for Software revisited.* Joenssu, FI, University of Joensuu, Dep.CSS, International Proceedings Series 6.

Fehlmann, T. M., 2016. *Managing Complexity - Uncover the Mysteries with Six Sigma Transfer Functions.* Berlin, Germany: Logos Press.

Fehlmann, T. M. & Kranich, E., 2012. *Using Six Sigma Transfer Functions for Analysing Customer's Voice.* Glasgow, UK, Strathclyde Institute for Operations Management.

Fehlmann, T. M. & Kranich, E., 2013. *Measuring Software Tests with COSMIC.* Kaiserslautern, Germany, DASMA e.V., pp. 93-100.

ISO 16355-1:2015, 2015. *ISO 16355-1:2015, 2015. Applications of Statistical and Related Methods to New Technology and Product Development Process - Part 1: General Principles and Perspectives of Quality Function Deployment (QFD), Geneva, Switzerland: ISO TC 69/SC 8/WG 2 N 14,* Geneva, Switzerland: ISO TC 69/SC 8/WG 2 N 14.