


# The Hybrid Model of Broken Agile Transformation in Big Telco Corporations

Dragan Stankovski<sup>1,2</sup><sup>a</sup>, Tsvyatko Dobrinov Bikov<sup>1</sup> and Dimitar Ivanov Radev<sup>1</sup>

<sup>1</sup>University of Telecommunications and Posts, Sofia, Bulgaria

<sup>2</sup>EPAM Systems, Sofia, Bulgaria

**Keywords:** Agile Transformation, Waterfall, Hybrid Model, Scrum Transformation, Telecommunication Transformation.

**Abstract:** The “Technical depth” it's one of the major challenges leading most of the project to delays or failures. Implementation of the Agile approach in its pure form does not fit do needs of the big corporations providing services in the telecommunications branch. This paper aims to present a hybrid model of “Broken Agile” that will accommodate and increase with a significant level the software delivery and development. The approach is resolving and providing a formula for the improvement of already working solutions in Agile projects.


## 1 INTRODUCTION

The pure form of Agile usually is a perfect match for small projects and companies who are delivering portions and pieces of software, mostly graphical user interfaces (GUI) and they are delivering continuously and often. In the world of big players and corporations especially those who are operating in the field of telecommunication, delivering software in production is quite complex driving hidden risks and big rocks that should be identified, monitored in real-time to have a successful build to production. Have been saying that from the other side there is also complex work from the back-end which nature cannot fit into the same timelines or sprints defined and dedicated for the graphical user interfaces execution. Back-end usually is hiding more risks and resolves problems, including testing it's more complex and it needs more time. Both pieces of this software need to be delivered in production in working shape without any issues and bugs, this challenge of small-time to deliver front end part and long term of the back-end is leading Agile to not sophisticated and this is why the new approach of a hybrid model is needed to close the 360 degrees of the life cycle in lines of software delivery process.

## 2 THE CURRENT ORGANIZATION OF AGILE IN THE BIG TELCO CORPORATIONS

The approach with back-end and front-end delivering of software is just one example but it's also a small part of the chain in delivering process for one big corporation like telcos. The huge and heavy organizations compile different structures and the processes for delivering have their dependencies from each other. The most useful example of such departments in big telco corporations are marketing, sales, solution, delivery, testing, infra, etc. (This is the most common structure the companies usually are using, and in rare cases, the structure can be slightly different in some of the departments can be merged or also can be split, up to the needs of the current business and of course up to the resources that are involved into development). All of them have their involvements and tasks in the process of delivering the pieces of software.

The regular flow usually is starting from the marketing perspective and PoV (Point of View) and it's pretty connected with the sales team to provide the best solution that can fit do the business itself. Even when those two departments are operating independently, they're very connected with the

<sup>a</sup> <https://orcid.org/0000-0002-0558-2236>

solution team who is the main player to analyse the current situation and the view of the road map for further development and expansions. These analyses are known as “AS-IS” and “TO-BE”. The structure mentioned above is facing two major issues - the first one is usually when marketing and sales provide and promise solution of product which currently it's not developed or it's not working according to the customer needs, and this is only because there is a gap in the knowledge of the product or gap between the communication in the Agile workflow and life cycle between all departments (M., Fowler, and J. Highsmith, 2001). The second challenge is related to the lack of analysis end business knowledge about the current situation and future map of the business solution.

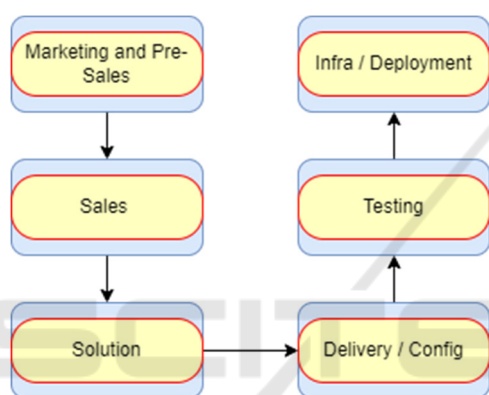


Figure 1: Common team organization in Agile structure and project delivering.

The challenges are continuing with implementation and execution that the delivery team needs to handle when handing the work from the solution team. In this case, a clear understanding of the creation of the content which is in the Epics and Features (Torado, 2019) levels and main responsibility of Product owns (O’Connell, 2013) should be described in a way that will be clear for developers to organize and finish their work between the sprints without any impact or leftovers. The next on the queue is the real challenge to find all possible ways of work to test and complete the execution of the test cycles with minimum issues end maximum numbers of executed test cases. Last but not least place especially GO, NO-GO which is the trigger to the infra team to go on production.

As is described above most of the teams have their impact on the process of delivering and the artifacts as Epics and Features, and User Stories need to be clearly defined and connected into a workflow with dedicated owners and of course timelines in which they need to be executed. Till now the idea to deliver

software in production is covered and answering the questions Who?, When?, and How? need to be delivered. Those 3 questions are the baseline on the structure in the real connection between the teams explained above. Therefore, in this paper, a new hybrid model is introduced to support and make stronger connections between teams avoiding or reducing dependencies and big rocks. In the next chapter with clear examples is explained how the teams need to be structured and how they need to communicate and manage their dependencies and tasks in different phases in the process of delivery.

### 3 CHALLENGES IN DELIVERY PROCESSES

The big telco corporations are facing a lot of issues and problems from different aspects, mostly they are related to the business understanding and lack of knowledge of the product road map and from the other hand, they've also related technical execution including processes that need to be established for the internal teams and third parties (3P). Described scenarios for the organization in the previous structure are the common between old and modern telcos providing services across the globe (O’Connell, 2014). In order to propose a suitable hybrid model that can increase and improve the process of delivering first need to identify what are the exact problems and how they can be resolved with our proposal.

- The biggest challenge for the telcos is the gap of knowledge in the structure and segmentation of the teams that need to be connected and work in Agile mode.
- The second but also very important problem is the ability and knowledge to identify, define and connect dependencies between different tasks that need involvement from different teams.
- Also, very important is to define the duration of the sprints and project increments for the product for delivery

Those three problems separately but also went together can impact the deliverables not only in the timelines but also in the quality of the code that needs to be live on production.

The next chapter will cover the proposal for a solution in one hybrid model that is using the two of the most used methodologies Agile and Waterfall in a combination of proper organization in the timelines, structure, and execution.

## 4 THE HYBRID MODEL OF AGILE DELIVERY AND ORGANIZATION STRUCTURE

As was shown in the previous chapters, one of the main reasons for delays of the projects in delivering on time a piece of software into the telcos industry is related to the accommodation of technical depth. This result is mainly of those three problems described in chapter 3. The end goal is to provide a final definition for one complex and a hybrid model that can decrease those deviations, in the following part separate description will provide challenges and solutions for the final shape of the suggested hybrid model of Agile and Waterfall's methodology highlighting the most common steaks in delivering projects. Figure 1 presents the common organizational structure and the main players including domains in a project lifecycle, as is shared, the problems are based on good skills to organize domains into a very good connection and chain in the life cycle so they cannot break the movement end execution of the software (Wanner, 2019). The first challenge is related to understanding the scope of work for each part so they can be grouped in one segment (this will reduce the dependencies between them). Figure 2 is presenting a new way of grouping the subdomains that can act more or less independently when it's come to solving complex end long-term tasks.

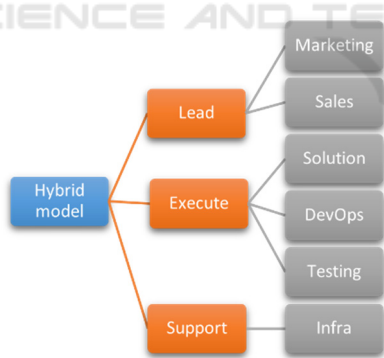


Figure 2: The new way of structure and grouping of domains with the new hybrid solution.

This solution of segmentation is providing three main options where the marketing and sales are grouped in one subdomain that is named lead segment. The main tasks are to communicate and provide clear direction, not only to the customer but also to the next part of the queue who will develop the product. This is leading to the second part which here is named as an execution part where the agreed and signed Epics and Features are moved to Development, Delivery, and Testing for real work. In

some structures of the big companies working in the field of telecommunications this part also is known as delivery but nowadays it's very common and fancy to use the terminology DevOps (Jennifer, 2016). This part also includes the major and critical decision makings in this structure which is called the Unit test and final GO decision to production. Last but not least part is connected with the support of all different activities across the team this is the main responsibility of the infra task and daily work.

The need for this segmentation in our proposal of the hybrid model is tidily connected with the organizational workflow on hand over the tasks. More detailed information is provided in Table 1., where the descriptions and rolls for each of those segments are presented and more detailed described.

Table 1: Hybrid organizational model.

New Sub-domain	The hybrid structure	
	Old structure	Main Roles
Lead	Sales and Marketing	Sales and Marketing Strategy and Roadmap
Execute	Delivery, Testing, Solution	Develop, Deliver and Test the product
Support	Infra / Deployment	Support Lead and Execute domains and own environments

### 4.1 First Break

The first break of Agile and place into the need of hybrid model exactly on those three levels. By the definition, all segments should work in an Agile mode, but in reality, marketing and sales cannot plan officially their work in the short term with sprints and product increments. The nature of this breaking is related to the fast market and huge competency that need to be a win and keep the place on a very huge and growing market. The worst-case scenario and unfortunately very often case is that in such segmentation and grouping sometimes sale promises “not existing” product with the only purpose to win the deal. Because of this reason the sales and marketing cannot work in the Agile framework so the combination of Agile and Waterfall is very acceptable.

This paper and the experimental results are based on real project implementation and execution in the period of September 2020 and June 2021 with 60 team members and participants split into the domains that are proposed in this hybrid model. With different proportions and combinations of agile and waterfall such as 10% waterfall 90% agile, 50% agile 50% waterfall, etc. most of the tasks are split and the

tracking marker is the technical dept. As a final result and the most valuable part of delivering and executing is presented into the combination of 20% Waterfall and 80% Agile – case 3 eq. Table 2.

Table 2: Definition and proportion of both methodologies.

Methodology	Case study and results		
	Agile	Waterfall	Technical Dept
Case 1	90%	10%	8%
Case 2	50%	50%	33%
Case 3	80%	20%	2%

The conclusion of the case studies in the approach and experiments is that the work of the leading squads in most cases can be dictated from the road map of the product end also the official licensing part who is already with agreed and familiar or fixed dates. Thus can be very easily scheduled and placed into the different sprints when need to be executed. The 20% of the waterfall are fulfilled with the task-related and small delays and changes related to human resources, new requirements, and specific things that arrived from the customer during the pre-sales conversations including the last minute changes from the governments and other third parties impacting documents and requirement changes of product.

The execution part is also attempting to work in Agile mode and the planning is reserved 100% for Sprint and PI (Project Increment) scheduled work.

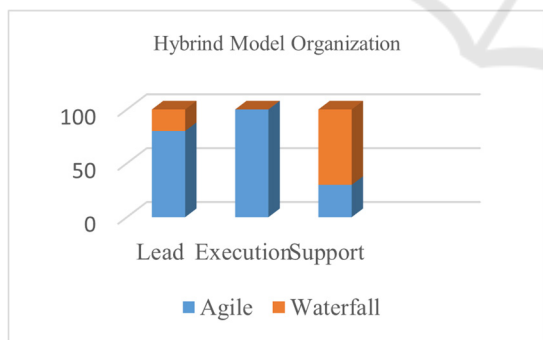


Figure 3: Results of the best approach with technical depts up to 5%.

The last group on the chain is named supportive because he is the less Agile organized and only 30% of their work can be properly planned as upgrades, installations, etc. The rest of the 70% is a pure waterfall and it is picked up with priorities to serve other departments.

## 4.2 Second Break

The next break of agile methodology is coming with the solution for the second problem related to the Sprint organization and time frames in delivery. This second issue is related to the difference and complexity between pieces that can combine pieces of software delivery - meaning front-end development is fast, graphical, and easily can be changed during the frameworks defined by the Agile community. On the other hand, the backbone or simply said back-end who is the major driver and execution of all requests arriving from the front-end part including the connectivity and storage into databases of changes and services it's very complex and slow to fit into the timelines and align in the same framework as front-end. Those two differences and complexity in development and delivering are also the reason why the big telcos cannot work in real Agile and already defined sprints and project increments (PI). The most common and already familiar way of structure for PIs is the direction of one month which includes 2  $\sum$  (sprints) each with a duration of two weeks (Maric, Tumbas, 2016). In the same experiment, it was slightly changed and conducted with the execution team or delivery team as the received results show the best synergy and velocity into delivering is when the sprints are organized well like it's presented in equation 1.

$$PI = \Sigma 1 + \Sigma 2 + \Sigma 3 \tag{1}$$

This means that one project increment for one software drop that needs to go on live in production should be with the duration of 8 weeks split into three different springs. The first two sprints will be with the duration of three weeks and the last one it's with only two weeks.

During our mini project, this proportion 3+3+2 was also tested because already known 2+2 is the most common way of working is unfortunately not good enough and doesn't fit the Backend and Frontend projects. With this, the technical depth of deliverables increased by eight times as is shown in Table 3.

Table 3: PI organization and delivery.

Approach	PI scope of Features (average)		
	Committed	Executed	Leftovers
2+2	10	6	4
3+3+2	20	19	1

This break and proposal for the hybrid model is the style of work where those sprints especially the first two sprints with the duration of a total of 6 weeks are working in pure Agile mode and the last one it's reserved and free to work in Waterfall - 2 Sprints can be handled and transfer to the latest Sprint. This "buffer" or "bucket" is also useful in showing good results because the same can be used for additional testing, handle different emergencies that are always coming at the last moment. Fig. 4.

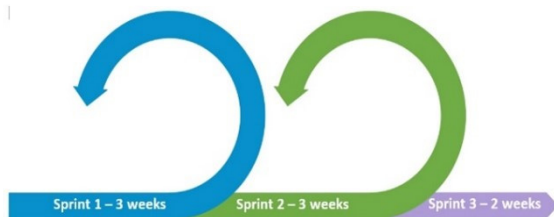


Figura 4: PI and Sprints organization in proposed Hybrid model.

Besides the support of those two breaks of the Agile methodology and provided modifications there is also one more problem to be solved to have a completed proposal for the hybrid model. In most of the projects, there is no differentiation between blockers and dependencies and continuously they're using the same terminology which is deviating the understanding of the urgencies and priority of deliverables. During project execution, there are different types of deviations and can be identified as outside of blockers' end dependencies. The different tasks in one project increment can be organized and easily can be linked with already familiar dependency structure:

- Finish-to-start (FS) - Task 2 can't start until task 1 is completed. The most common type of dependency.
- Start-to-finish (SF) - Task 1 can't finish until task 2 is started. The least common type of dependency.
- Start-to-start (SS) - Task 2 can't start until task 1 has started, but task 1 does not have to be completed before task 2 can begin.
- Finish-to-finish (FF) - Task 2 can't finish until task 1 is completed (Sara, 2020).

Mentioned attributes of deviations can be monitored and tracked into the different tools across the organization in the timeframes that the project need to accommodate, but the biggest challenge is to have only one task or artifact that can define end measure all of those dependencies, assumptions questions, links between themselves in one Risk assessment. New Artifact that will combine old deviations and measurements of the risk during the

delivery is defined as "Initiative". The final goal is to provide feedback for the current state or bring to escalation if the execution is delayed or it will fail. This is also known as the risk assessment and therefore the owner by defining the three different levels on track, needs attention and skill to orchestrate the risk including updating the feedback for the current state of all dependencies and deviations. It is already known not only for the Agile, but also for our hybrid model: "The goal of Agile teams shouldn't be to eliminate dependencies entirely, but to reduce complexity, improve flow, and increase their ability to predict how dependencies will impact their ability to deliver work." (Brook, 2020). Exclusions are possible in rare cases where the structure is very large there is also a need for one more level of monitoring and managing, this level can be on product or project level with the same marks as milestones.

This is why with this proposal of the hybrid model there is no aim to reduce the dependencies they are a good path for the correct organization and it's showing the clear vision and the road map of developing of the product in the same time the most important with the Initiatives is to set the risk assessment and prevent from the big rocks, possible delays caused by the technical depth and possible cases of rollback after official upgrades to production.

## 5 CONCLUSION

The Hybrid model proposed in the chapters above is introduced in the real work environment and the real project implementation with the duration of nine months. Total participants, an average of 60 engineers such as developers, testers, salespeople, solution architects and managers, support engineers. All of them were divided into six streams with an average of ten people. Before implementing and modifying the agile methodology of the Hybrid model the velocity of the delivering it's measured to 65% delivered compared to planned work to be done. The rest of 35% it's mainly related to the increased technical depth and in some cases, the tasks and leftovers move between more than six sprints. In the total of four official releases, 1 is a complete rollback. Reporting hours of the engineers over exceed the normal working time related to catching up and delivering releases on time. The measurements register an average of 50 working hours per week, which is 25% overtime of the regular.

For the same timeframes - September 2020 till June 2021 with the same capacity and implementation of the hybrid model presented in this paper the final

results increase not only the level of delivery without failures but also the velocity. A percentage between committed and executed increased to 95% which is 30% more than the pure agile work style. The reason for the five percent is related to the aggressive market and changes that need to be implemented at the last minute. The working hours of the team members decrease to an average of 42 hours per week which is a significant improvement. Within this total working, time is reported not only as real working hours but also as a time for personal development and self-learning (Stankovski, 2022). In conclusion, the major problems described in chapter 2 and proper implementation of the proposed hybrid model between agile and waterfall schedule in the modified time frames and sprints is increasing the deliverables of the final software version in production and comparing with the previous style of work without rollbacks.

## ACKNOWLEDGEMENTS

Terminologies mentioned in this paper as GO, and NO-GO in the real work environments meaning approval (GO) from the stakeholder and final decision to move the product to production or stop upgrades if the acceptance criteria are not covered (NO-GO). Terminology for AS-IS presents the current situation and environment that the company is following and working. TO-BE is a suggestion of how they should transform or work in order to achieve better results.

If any should be placed before the references section without numbering.

## REFERENCES

- Brook, A. (2020). Agile Teams: Dependency Management and Visualization
- Fowler, M., Highsmith, J. (2001). The Agile Manifesto, Software Development, Vol. 9, No. 8, pp. 28-32.
- Jennifer, D., Katherine, D. (2016) Effective DevOps, Building A Culture Of Collaboration, Affinity, And Tooling At Scale, O'Reilly
- Maric, M., Tumbas, P. (2016) The Role of the Software Architect in Agile Development Processes, Strategic Management, Vol. 21
- O'Connell, T. (2013). The Scrum Product Owner – Customer Collaboration & Prioritizing Requirements, in Proc. ICSEA13
- O'Connell, T., (2014). An Analysis of the Implementation of Agile Software Development Practice in Irish Industry, International Journal on Advances in

- Software, vol 7 no 3 & 4, <http://www.iariajournals.org/software/>
- Santos A., Kroll J., Sales A., Fernandes P. and Wildt D. (2016). Investigating the Adoption of Agile Practices in Mobile Application Development. In Proceedings of the 18th International Conference on Enterprise Information Systems - Volume 1: ICEIS, ISBN 978-989-758-187-8, pages 490-497. DOI: 10.5220/0005835404900497
- Sara, M. (2019) Dependency Explained (with Examples) <https://www.tacticalprojectmanager.com/finish-to-start-dependency> <https://www.planview.com/resources/guide/what-is-agile-program-management/agile-teams-dependency-management-visualization/>
- Stankovski, D., (2022). The Measurements that matter with hybrid model of estimation, The Journal of CIEES, <http://journal.ciees.eu/index.php/ojs/authorDashboard/submission/12>
- Torado, D. (2019) “The Epic Guide to Agile: More Business Value on a Predictable Schedule with Scrum”, pp. 26-67
- Wanner, R. (2019). How to successfully apply Agile project management and Scum, Proconis, First Edition ISBN: 978-1983653995 V1.0