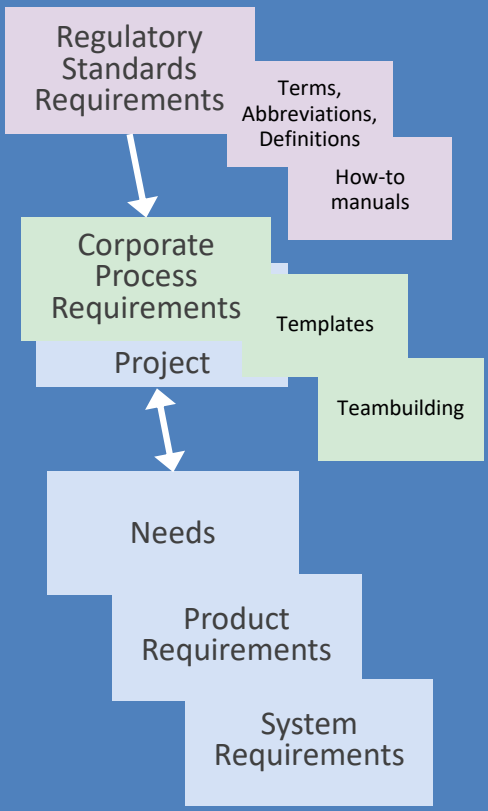
PaR methodical framework

Processes as Requirements

The Booklet



Systematic
Software
Engineering

PaR – Processes as Requirements
The Booklet
methodical framework

Version: 01.January.2021
1st printing 01.January.2021
ISBN 978-3-753141-74-9

Publisher: Ralf Bürger
Wilhelmstraße 24
45527 Hattingen
Germany (Europe)
eMail@RalfBuerger.de

Print: epubli – a service of neopubli GmbH, Berlin

Community: https://ProcessesasRequirements.info

Acknowledgements: To the people and companies that support me:
Sascha Kobus & Frank Zielke (KoDeCs)
Martin Becker & Max Schmitt (Fraunhofer IESE)
Bernhard Doleschel (ease solutions)

"First, don't be afraid. ...
Second, do what you think is right. ...
Finally, build a community.
No one does big things by themselves."
(President Obama, 18.May.2020)

PaR excellence

ProcessesasRequirements.info

your challenges / PaR solutions

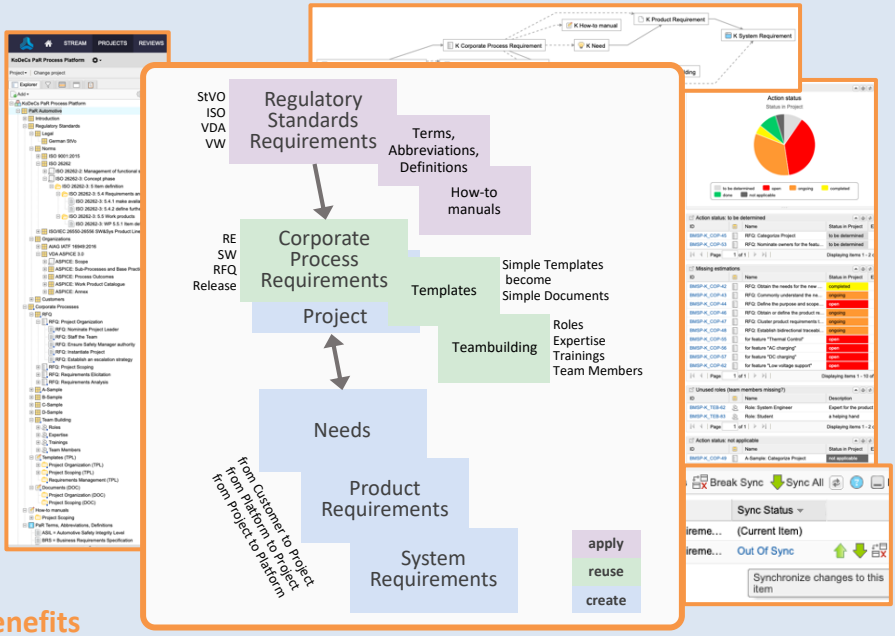
1 / ***help*** **all types of projects with flexible corporate development processes**
1 **PaR:** define and reuse the processes as requirement sets in your RE tool

2 / ***merge*** **regulatory standards with corporate development processes**
2 **PaR:** define also the standards as requirements and add traceability

3 / ***learn*** **with the project teams by established sustainable processes**
3 **PaR:** unite process and product requirements, but improve both

4 / ***comply*** **continuously with processes and standards in the projects**
4 **PaR:** use the features of your RE tool for bi-directional traceability

5 / ***monitor*** **actual project and product maturity progress**
5 **PaR:** measure the status of all requirements, documents and reviews



benefits

① lightweight efficient processes are welcomed by developers
② uniting “what” and “how” in the teams’ tools is more agile
③ standards and processes are focusing on projects and learning culture
④ project teams are empowered to self-organize the compliance
⑤ step by step, teams apply flexible platform techniques also for processes
⑥ it’s a systematic holistic methodical framework that is easy to adopt and adapt
⑦ it makes true transparency for actual process and product maturity

Table of contents

PARADE OF CHALLENGES SHOWING THE NEEDS 4

How do you design corporate development processes that are flexible enough to be a *HELP* for all types of your projects? 4

How do you *MERGE* all needed regulatory standards with the corporate development processes to make it a holistic approach? 4

How do you establish sustainable corporate development processes that can *LEARN* together with or from the project teams? 5

How do you ensure that your projects continuously *COMPLY* with corporate development processes and regulatory standards? 5

How do you *MONITOR* the actual progress of the project and product maturity, in addition to monitoring time and budget? 5

PARIS (PAR INFORMATION SYSTEM) 6

PaRis detailed 6

PaRis overview 9

PaRis abstract 9

PARTOUT - TOOL FEATURES TO SATISFY THE NEEDS OF PAR 10

Feature 1: Definition of requirement item types 10

Feature 2: Implementation of the PaRis map 10

Feature 3: Evaluation of project maturity 11

Feature 4: Compliance checks by standards coverage 11

Feature 5: Support for process versions 11

Feature 6: Reuse of requirements sets 11

Feature 7: Synchronization of requirements sets 11

Feature 8: Definition and management of variability 11

PARAMETERS TO MEASURE ANYTHING 12

KPI 1: "Are we still on time?" 12

KPI 2: "Are we on budget?" 12

KPI 3: "How much is the coverage?" 12

KPI 4: "Are we compliant?" 12

KPI 5: "What is the maturity?" 12

PARACHUTE FOR PROJECT PLANNING 13

PARADIGM SHIFT 14

PaRade of challenges showing the needs

I have identified **5 challenges** from my coaching over the past few years.

Let's examine the **challenges** and derive some **needs** for a methodical framework that might at least mitigate these challenges.

1 **How do you design corporate development processes that are flexible enough to be a *help* for all types of your projects?**

Accepting this challenge is important for your small projects with a co-located team as well as huge multi-year projects with distributed teams.

Flexibility is also important for offering help for different types of projects, e.g. for research, for acquisition, for prototyping, for series production.

1 **Define the Processes as Requirements in dedicated sets that can be reused as needed in the projects applying the teams' standard RE tool!**

Organize the requirements sets as a process platform that can be continuously improved, also by the project teams – from teams for teams.

The projects can reuse "how" (the process) they work, according to "what" (the product) they shall do, rightsizing the processes.

A standardized process platform eases to bring new people into the projects, rotate people and reuse any artifacts between projects.

2 **How do you *merge* all needed regulatory standards with the corporate development processes to make it a holistic approach?**

Accepting this challenge is important because regulatory standards are needed, e.g. to ultimately value people's lives higher than business profits.

Enthusiastic developers often focus on solving the technical problem only, forgetting corporate sustainability, e.g. for market callbacks ages later.

2 **Define the regulatory standards also as requirements and relate them to the corporate development processes with bi-directional traceability!**

Link standards to projects, because standards give teams the basic experience and are not made for auditors or assessors.

"Duty" (no options) and "Freestyle" (doing it my way) gets merged by relating the regulatory standards to the corporate processes in the RE tool.

How do you establish sustainable corporate development processes that can *learn* together with or from the project teams? 3

Inspecting and adapting often, learning quickly, improving continuously are obligations for professionals, but challenges for company culture.

Unite the process requirements with the product requirements in the projects, but improve both platforms by synchronizing requirements! 3

Mainly the self-organized agile teams will like the approach to improve the processes from the projects for the projects, based on Retrospectives.

Product excellence may be specific for each project, but methodical excellence can be learned and reused from project team to project team.

How do you ensure that your projects continuously *comply* with corporate development processes and regulatory standards? 4

When projects apply the corporate processes but still do not comply to the standards then the compliance of the process itself is the challenge.

When the standards and/or processes are pulled as requirements into the projects, the RE tool can ensure the bi-directional traceability! 4

With good modern RE tools, a dashboard makes missing or broken traceability transparent through diagrams, filters and lists.

With bi-directional traceability compliance is monitored automatically and continuously instead of assessed manually once in a while.

How do you *monitor* the actual progress of the project and product maturity, in addition to monitoring time and budget? 5

Traditional project planning and monitoring by tools, project managers and controllers often fails because measuring maturity is a real challenge.

Defining a status for process requirements, adding templates for simple documents, and real reviews will help measuring progress in maturity! 5

The united process and product requirements create a complete picture of the project and support real planning and measuring.

Adding effort estimation and status to the united requirements sets further improves the project transparency, valuing openness.

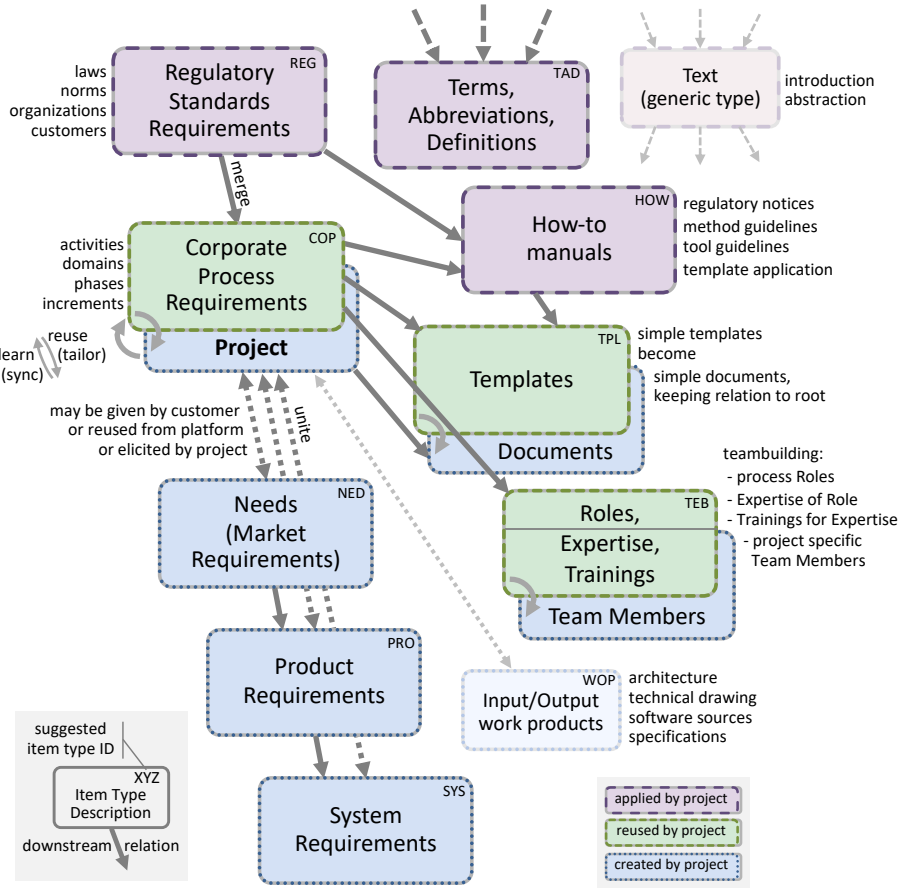
PaRis (PaR information system)

The solution path from the challenges via requirements to the resulting information system is described in The Book, also with more diagrams.

PaRis detailed

The detailed map shows the requirement item types and their relations.

- Some requirements are simply applied by projects without change, for detailed lookup or guiding help.
- Other requirements are rather inputs to be reused by projects, also to be modified or extended.
- Reused items, including certain work products, finally become items that are created by projects.



The following explanations relate to the items and arrows of the **PaRis** map, as well as to the Needs ❶ derived from the Challenges ❶. Furthermore, they are ordered according to the Benefits ① named on page 2 (the one-pager **PaR excellence**).

The core idea of **PaR** is that the predefined sets of Corporate Process Requirements COP are reused (pulled with traceability) in the projects to make them become Project activities, planned with project parameters. ❶

Accordingly, the corporate process Templates TPL are reused and filled to become simple Documents as project results and evidence. ❺

Also, the Roles TEB (with related Expertise and Trainings) are applied by assigning actual project Team Members.

This simple "adopt & adapt" approach makes lightweight efficient processes that are welcomed by project developers because the flexibility of the framework makes them a real ***help*** for all types of projects. ① ❶

Defining the Regulatory Standards REG and the additional standards for HOW and TAD also as requirements and relating the COP and TPL to them is a good way to ***merge*** all prescriptive project aspects consistently. ❷

The downstream item is always created from the upstream item. From REG to COP it is a "derive" (*merge*) action, from COP to COP (*Project*) a "reuse" action and from COP to PRO a "satisfy" (*unite*) action.

Within the product family the PRO are derived from NED, then further to SYS and the domain requirements DOM. The DOM doesn't show up in the **PaRis** but is discussed in detail in **The Book** when it comes to OMG®[1] UML®[2] diagrams and SysML™[3] modeled process architectures. Finally, the NED, PRO, SYS, DOM ("what") may be united with the COP ("how") to become a truly holistic agile product family model. Then the teams ***learn*** in projects and improve the product platforms systematically. ❸

Uniting "what" and "how" on a feature level can make good Epics for a Backlog, as used in agile approaches and also scaled agile approaches. Defining this work-to-be-done in sets of requirements is one way to apply agile mindsets in large-scale traditional projects. Refining the Epics into Stories can still keep the idea of combining "what" and "how". When it gets to Tasks in the Sprint Planning the "how" gives an idea of the methodical doing in the Task and the "what" may be the input or the resulting output of the Task. ② ❺

---

[1] see https://www.omg.org (Object Management Group®)
[2] see https://www.uml.org (Unified Modeling Language™)
[3] see http://www.omgsysml.org

The explanations above already show how the **PaR** framework focuses the standards and processes on the project teams, because they are made for them and not for assessors, auditors or central departments. I like to quote my favorite saying: "*Processes are reuse of best practices*". Therefore, the processes should be improved from the project teams to ***learn*** for the next project teams. This learning culture and can be technically supported by synchronizing the projects' improved activity requirements back into the processes, e.g. after agile Retrospectives. This should be assisted by experts from central departments, as they know the entire corporate project portfolio and all product platforms.

It is good advice to keep a technical relation to the original item when creating COP, TPL and TEB requirements by reuse from standard to project. This technically enables efficient learning from a project for the standard by tools' synchronization features. And it also creates bi-directional traceability that helps the project teams to ***comply*** continuously with processes and standards. The checks are just a click away in the tool, or they even can show up automatically in a dashboard at tool start. Of course, the tools must provide these features, and we come back to the tool discussion in a later chapter. Given these features in the tools, the **PaR** framework enables the project teams to self-organize measuring coverage and compliance continuously.

The dotted arrows are alternatives depending on the project purpose and the tool infrastructure. NED, PRO and SYS may be inputs as well as outputs. Larger work-products (WOP) may be related as in- and outputs depending on the tool capabilities. They show evidences for compliance also. Applying **PaR** in a dedicated project will make these arrows solid ones. Project teams that work in or upon product platforms know how to play these games and are motivated by **PaR** to apply these approaches step by step also for processes, making the COP hierarchy in fact becoming a process platform. **The Book** discusses product and process families as well as product and process platforms, also to unite both to a true holistic agile product family approach.

The dashed arrows enable to derive terms, abbreviations and definitions by the TAD from any other item type. This ensures that all people in the company speak the same domain language.

The "Text" item type in the **PaRis** is generic and may be used when no dedicated item type shall be created. It may represent a requirements folder, introductions or specific explanations. This item type may be called differently in the tools, but usually something generic exists.

The TEB may be broken down into four dedicated item types (ROL, EXP, TRA and TEM), if the organization shall be shown with more details in the RE tool. Otherwise, the identification of the dedicated purpose can be given by naming convention in the TEB items. Keeping the organization definition close to process and product requirements encourages the teams to review the activities often. ❺

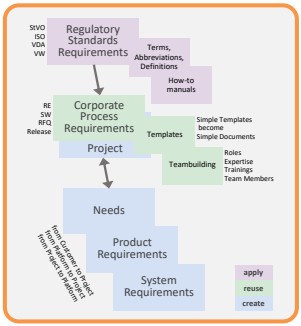
PaR brings regulatory standards, corporate processes, functional and technical product aspects, simple templates for simple documents, and also team organizational aspects all into the RE developer tool, because all these are requirements to the project teams "how" to do the "what". This simple idea makes **PaR** a systematic holistic methodical framework. ⑥

Since **PaR** uses a tool that is applied and well known by the teams anyway, the framework is also easy to adopt and adapt by the teams for their daily work. And due to the bi-directional traceability it also delivers true transparency to ***monitor*** the actual progress of process and product maturity in the project in an intrinsic fashion. We come back to the measurement discussion in a later chapter. ⑦

PaRis overview

The overview diagram is hiding many details. It is good for an entry to the discussion or for a management view.

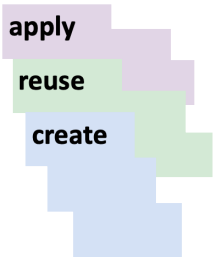
We use it also for **The Slide**, **The Deck**, **The Page**, for the cover of this **The Booklet** and for the onepager at the start of this **The Booklet** and **The Book**.



PaRis abstract

This small diagram further reduces the **PaRis** to an abstract level (rather some artwork), used for posters, websites, presentation slides or as logo.

It still shows that requirements are applied, reused and created. Also, we still see that all three ways make up hierarchies due to the typical domain and project complexity that we address with **PaR**.



PaRtout - Tool features to satisfy the needs of PaR

This set of features defines what a requirements management tool should bring in to be able to fully support the PaR methodical framework. See the special exemplary implementation booklets also, e.g. PaR – for Jama.

Converting processes from a rather graphical process design tool to a requirements management tool can be yet another challenge. When the processes are modeled in a database, all information is available but maybe not exported. When the processes are rather drawn it can get really difficult. At least an export to PDF or word processor should be possible, but then further conversion is needed to import, and the relations get lost anyway. A stepwise or partial conversion may make sense, e.g. first for the roles, the templates and the core process "software unit design" as an agile PaR MVP[4]. Team and management acceptance can thereby be tested early, and it may become a helpful success story supporting the next steps.

The stepwise conversion also helps considering using both graphical and requirements approaches in **parallel**. A graphical process flow visualization may be fully sufficient for business processes and production processes and may help for a good overview and understanding of the process landscape of complex development projects. However, when it comes to working within the process in the very individual project, it is more flexible to have the Processes as Requirements in the developers' tool. Even if a process design tool offers project instantiation features, it is still far away from the needed flexibility and from agile approaches for holistic product family models.

Feature 1: Definition of requirement item types

Bringing it all into the requirements management tool means managing many different aspects. This can be done with a super-set of attributes, but it makes more sense to define different item types for the different purposes, each having specific attributes and corresponding form fields.

Feature 2: Implementation of the PaRis map

It should be possible to implement the PaRis map by defining a dedicated relationship model for the defined item types. If different item types cannot be defined in the tool, then at least basic relations should be possible, because this is also required for bi-directional traceability in standard requirements management. Relating by hyperlinks (http) or naming convention (like "see xxx") is **not** sufficient.

[4] MVP = Minimum Viable Product for starting the evolution.

Feature 3: Evaluation of project maturity

When we define Processes as Requirements, then monitoring the project progress within the requirements management tool is mandatory. This can be done by a field or attribute for a status definition, or by analysis from outside with other tools. The deduction of documents from standard templates and relation to work-products should also be checked by the tool. Estimations and current effort for time and money should be added as calculated fields or attributes for analyzing actual progress.

Feature 4: Compliance checks by standards coverage

Most projects are subject to some mandatory regulatory restrictions. Therefore, checking for coverage and for compliance is necessary. This can be a trace view function within the tool, ideally with additional analysis options, or it is done by external analysis of exported data.

Feature 5: Support for process versions

It should be possible to define entire processes or parts of them as baselines. These should lead to reusable versions, also to enable permanent consistency.

Feature 6: Reuse of requirements sets

To be able to apply the Processes as Requirements to multiple different projects, a reuse functionality must be available. The worst case is "copy & paste", better is sharing, the best case may be reusing a dedicated version, maybe based on branching/merging concepts or on binding variability to variants.

Feature 7: Synchronization of requirements sets

Later process improvements should be synchronized into the projects, and improvements by the project teams should be synchronized back into the process standard. This requires dedicated tool functionality, e.g. “update”, “merge” or “synchronize”.

Feature 8: Definition and management of variability

The good experiences from Product Family Engineering should be applied to process landscapes to define project specific process variants from a variable process platform. Therefore, it must be possible to design variability in process requirements and then reuse dedicated variants in the projects by binding the variability.

PaRameters to measure anything

Time flies! Inspecting and adapting frequently is important to stay focused on the actual goals. We should highly value collaborating with the customer. We should make product increment reviews with the team and the stakeholders after every iteration. In addition, we should continuously measure automatically to avoid spending too much time for formal inspection. Find the detailed PaR**ameter** descriptions in The Book, based on a systematic approach for measuring KPIs (**K**ey **P**erformance **I**ndicator).

KPI 1: "Are we still on time?"

Add fields to the requirements items of the corporate process for the "estimated" and "done" timely effort. Also plan the corporate process requirements in the project on a target (milestone, release, gate, Sprint, etc.). Use these parameters for continuous measurement.

KPI 2: "Are we on budget?"

Add fields to the requirements items of the corporate process for the "needed" and "consumed" budget. Use these parameters - together with the timing - for regular measurement (in projects with subsequent production, time is often more critical than budget).

KPI 3: "How much is the coverage?"

Use bi-directional traceability to measure how many process requirements are being applied and in which status they are in the project. Measure the coverage of the regulatory standards accordingly. Use these parameters for continuous measurement.

KPI 4: "Are we compliant?"

Use bi-directional traceability between regulatory standards requirements and corporate process requirements to measure potential project compliance. Use bi-directional traceability between corporate process requirements and reuse in projects to measure actual project compliance. Do the same for work products to show evidences.

KPI 5: "What is the maturity?"

Measure the degree of unification of process and product requirements, which may be the degree of application of process and product platforms. Rate it in relation to the level of customer satisfaction and in relation to the project progress discussed above. Also check for documents and other larger work-products.

PaRachute for project planning

We have learned over many years that **traditional project management** often is not very successful. Reasons include the increasing volatility of the environments and the ever-shorter improvement cycles. These changes challenge classical upfront planning of the projects. **Agile approaches** within the projects' boundary conditions are therefore becoming more and more popular and prove that they can be more successful. Agile approaches pass more organization and responsibility to the teams, gain as much transparency as possible, and inspect and adapt often.

In **traditional project management** we answered with Gantt-Charts for an upfront planning like rolling waves. Meanwhile some project managers thought it is a good idea to take the process as template for the project schedule, but it was insufficient when for example multiple software features had to be implemented. Therefore, others took the requirements as input for the project planning, but then the activities were missing for the estimations and role assignments. **Agile approaches** break down Backlog Epics into Stories and further into Tasks to figure out all the work that is to be done and thereby offer a more intrinsic and natural way to plan the project.

The **PaR** framework takes all of this into account by uniting product requirements with **P**rocesses **a**s **R**equirements in the project in a traceable manner in order to plan them in phases or gates or Sprints or releases or samples, or however the stuff is to be delivered iteratively, in order to get early and frequent feedback. This framework respects the traditional rolling wave planning, the long-term milestones for the corporate controlling, the product requirements as well as the standards and processes, and it also suggests to self-organize in an agile fashion to frequently refine Epics and Stories in Backlogs.

In the **requirements management tool,** we finally get a hierarchy of requirements sets. On the last level of this hierarchy, we find the items that are meant to be executed somehow in the given order, while the sets themselves often can be planned for parallel execution. Every topic has some natural order of its parts, but on the other hand even complete project phases may overlap in real life.

PaR respects the ideas of traditionally designed processes, but also supports the needed flexibility for agile teams to efficiently work the stuff off in the projects. The **PaR** framework can thus become your **PaRachute** for struggling project planning, organization and measurement.

PaRadigm shift

All I ask is to see regulatory standards and processes for what they are: requirements "how" the "what" shall be done. This enables teams to quickly take on more responsibility and decide when which process requirements shall be adopted and adapted as needed[5]. Thereby, **PaR** puts the project teams in the driver's seat of process evolution within their short project life cycle, while the central process designers really need to take on long-term responsibility for process maturity and standardization as a service. This reduces processes and planning to the maximum. ;-)

If you always do what you've always done,
you'll always get what you've always got. *(Henry Ford)*

Not everything is completely new with **PaR**: We have requirements as always and manage them in the same tool as always. The content of standards and processes, how-to manuals and definitions, templates and roles is not new either. On the other hand, the way we manage the content is completely different, and that makes it a paradigm shift at the framework level.

The electric light did not come
from the continuous improvement of candles. *(Oren Harari)*

Following a paradigm shift means thinking anew. Thinking anew at first does ***not*** mean a transition of the old as a whole. Instead, **PaR** offers a new methodical framework in which many fragments of the old framework can be rearranged and reworked. But it still requires a change in our attitude.

Everyone thinks of changing the world,
but no one thinks of changing himself. *(Leo Tolstoy)*

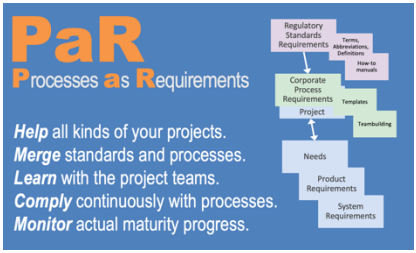
The **PaR** framework can also only be partially implemented and varied greatly. The more ideas one can pick, the better it is for the project teams. And doing it in steps sounds good too. Get started and learn, because success stories are often required to convince management and certain departments of a new methodical framework that actually represents a **PaRadigm shift**.

---

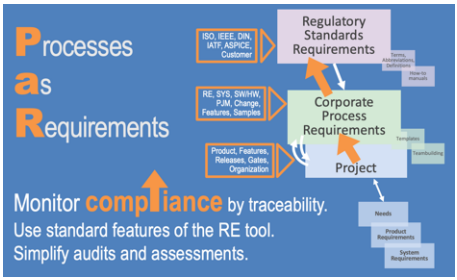
[5] Read "Two Types of Authority Leaders Must Give to Self-Organizing Teams" (Mike Cohn) https://www.mountaingoatsoftware.com/blog/preview/1680

see http://ProAsReq.info/splashes.htm for larger diagrams

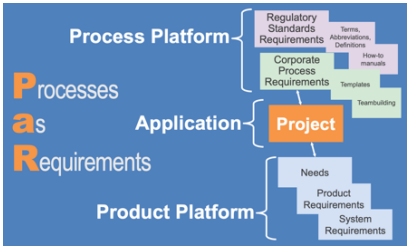
Intro Processes should be an exoskeleton that helps the team to move safely and efficiently through the storms of the project. When your processes rather become a heavy backpack that a project must carry in addition, then designing the Processes as Requirements may be your solution.



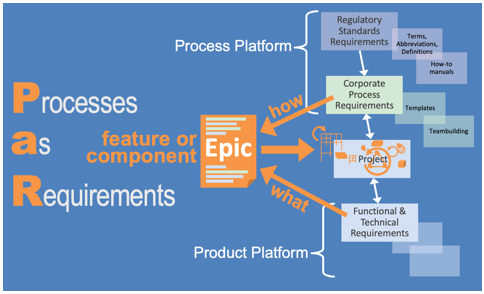
Compliance With PaR compliance can be continuously monitored by bi-directional traceability, using the standard features of your Requirements Engineering tool. This simplifies or even saves audits and assessments.



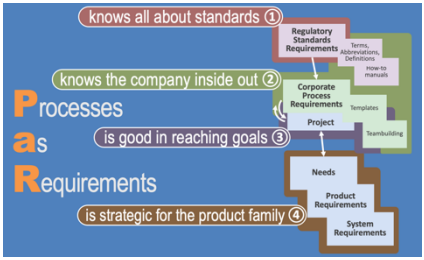
Platforms With PaR variability management can create a process platform from these process requirements, just like the product requirements of your product platform. The projects apply both in a united fashion, solving an old problem of project planners also.



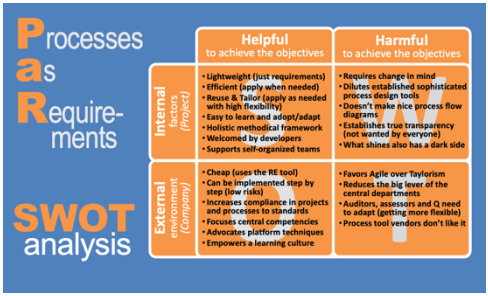
Agile With PaR uniting "what" and "how" on a feature or component level makes good Epics for the Backlogs of scaled agile approaches.



Teams With PaR you can organize in teams that really work together. While the teams focus on their objective the members can switch from team to team to work in a cross-functional fashion.



SWOT When you design Processes as Requirements you should know the good benefits that you get and the few risks that you take!



PaRdon – there is more

If you sense some urgency now to change culture, processes or tools in your company, organization or projects, then you have already taken the first of the eight steps that lead to change, according to John Kotter.[6]

It's up to you now to take the next steps. PaRticipate in our training & coaching, benefit from our business- and tool-experience, from our out-of-the-box requirements sets, and from our stuff below. It may help you to further distribute the ideas in your company, organization or projects, either as PDF or "on paper" in high quality.

For personal assistance please don't hesitate to contact me
Ralf.Buerger@ProcessesAsRequirements.info
and the community at
ProcessesAsRequirements.info (ProAsReq.info).

The Book The Slide The Deck The Page for *<tool>*

One more thing[7]: The Mug ;-)

[6] Read "Our Iceberg is Melting: Changing and Succeeding Under Any Conditions" (Kotter, Rathgeber) – ISBN 978-0399563911

[7] 3 legendary words that Steve Jobs used to announce something new (YouTube video: https://www.youtube.com/watch?v=cO-2NAl7Sm0).

The Book has many more good references.

… From the graphical representation in 3 levels we can derive an organizational responsibility that is not that visible today in many of our departments. Setting up the regulatory standards – i.e. the boundary conditions – separately from the processes and then relating it to each other is something we don't do that clear today. But that would be much more efficient also for sure. …

Central process department, a German automotive supplier

… I'm convinced that PaR is the next step to be more efficient and agile in projects even though you have to fulfil A-SPICE, ISO 26262 and ISO 21434. …

Sascha Kobus, CEO KoDeCs GmbH

… Very promising approach, which exploits the reuse potential for product and process aspects in a unified manner. …

Dr. Martin Becker, Department Head Embedded Systems Engineering at Fraunhofer Institute for Experimental Software Engineering (IESE)

… The variant development process for architecture is among the best we have seen. We consider this approach to be state of the art and benchmark. Especially the strong link between platform and project architecture …

Feedback from an expert discussion of a process for platform-based product development, that I created over some years for a German automotive supplier. My basic platform ideas of that process also made their way into the **PaR** framework for process platforms.

Did your processes become a heavyweight backpack to be carried by the projects, rather than a lightweight intrinsic approach that really helps the teams to navigate through the storms of the projects?

It gets better when you design regulatory standards and Processes as Requirements that are reused and improved by the projects.

Estimated complete reading time: 2 hours
Estimated complete application time: 2 years ☺

Ralf.Buerger@ProcessesAsRequirements.info
https://ProcessesAsRequirements.info