Uso de Métricas en Producción de SW. a Escala

Delivery metrics

OpenSouthCode 2023
2023-06-10
Málaga. Spain

Agustín Benito Bethencourt
Independent consultant

Table of Content

1.– The speaker

Agustín Benito Bethencourt

@toscalix

http://www.toscalix.com

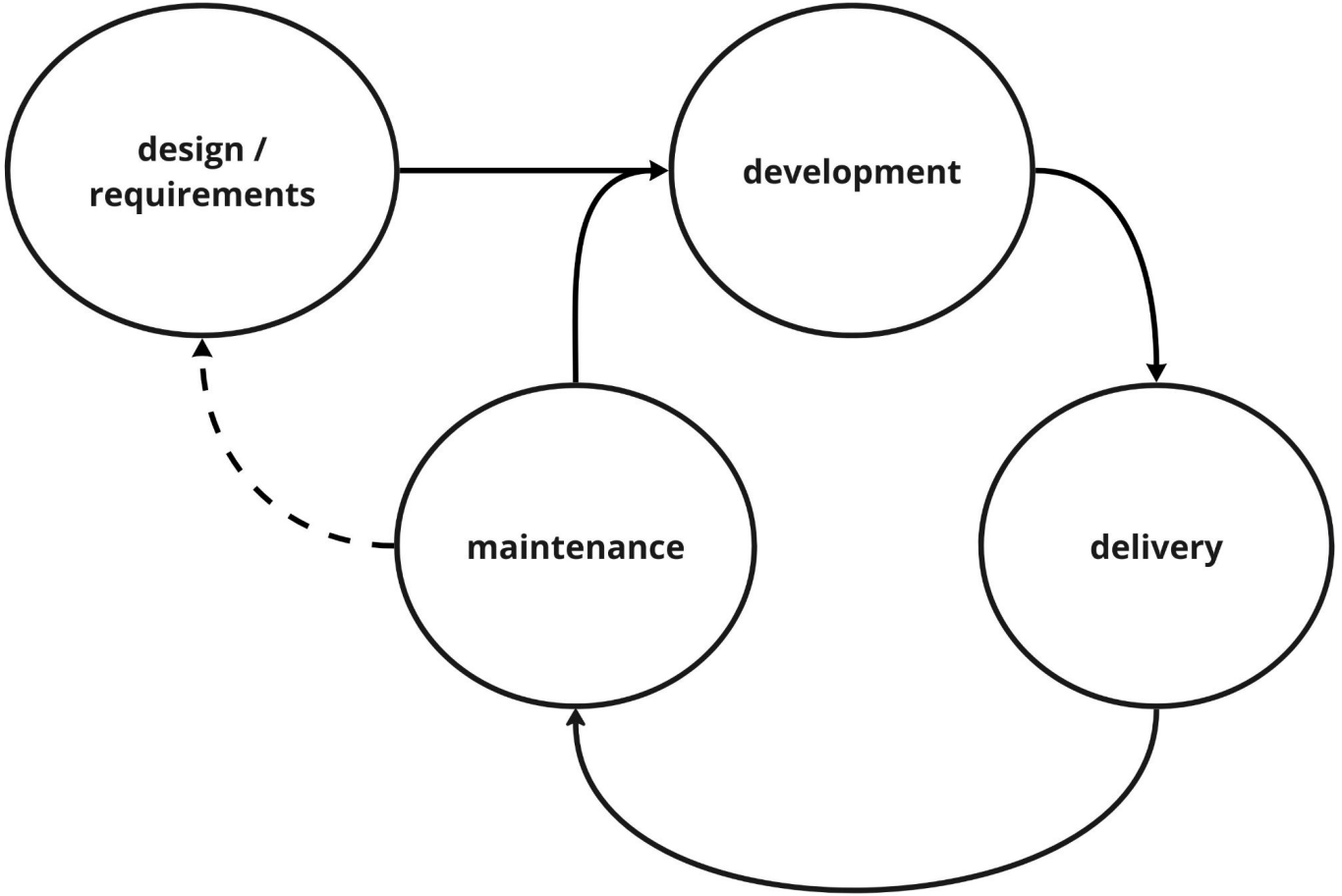
- Currently independent consultant, helping companies in the design and delivery of SW products.
- FLOSS, agility, Continuous Delivery and remote work advocate.
- Worked in different industries, mostly focused in FLOSS platforms and distributions (OSs).
- Worked for SUSE, Linaro, MBition (Mercedes Benz), Eclipse Foundation, Codethink, etc.
- In FLOSS since 2003. Contributor to numerous projects. KDE eV and KDE España member.
- Based between Málaga & Canary Islands, ES.
- Slimbook + openSUSE + KDE user.

If you can't explain something in simple terms, you don't understand it. - Richard Faymann

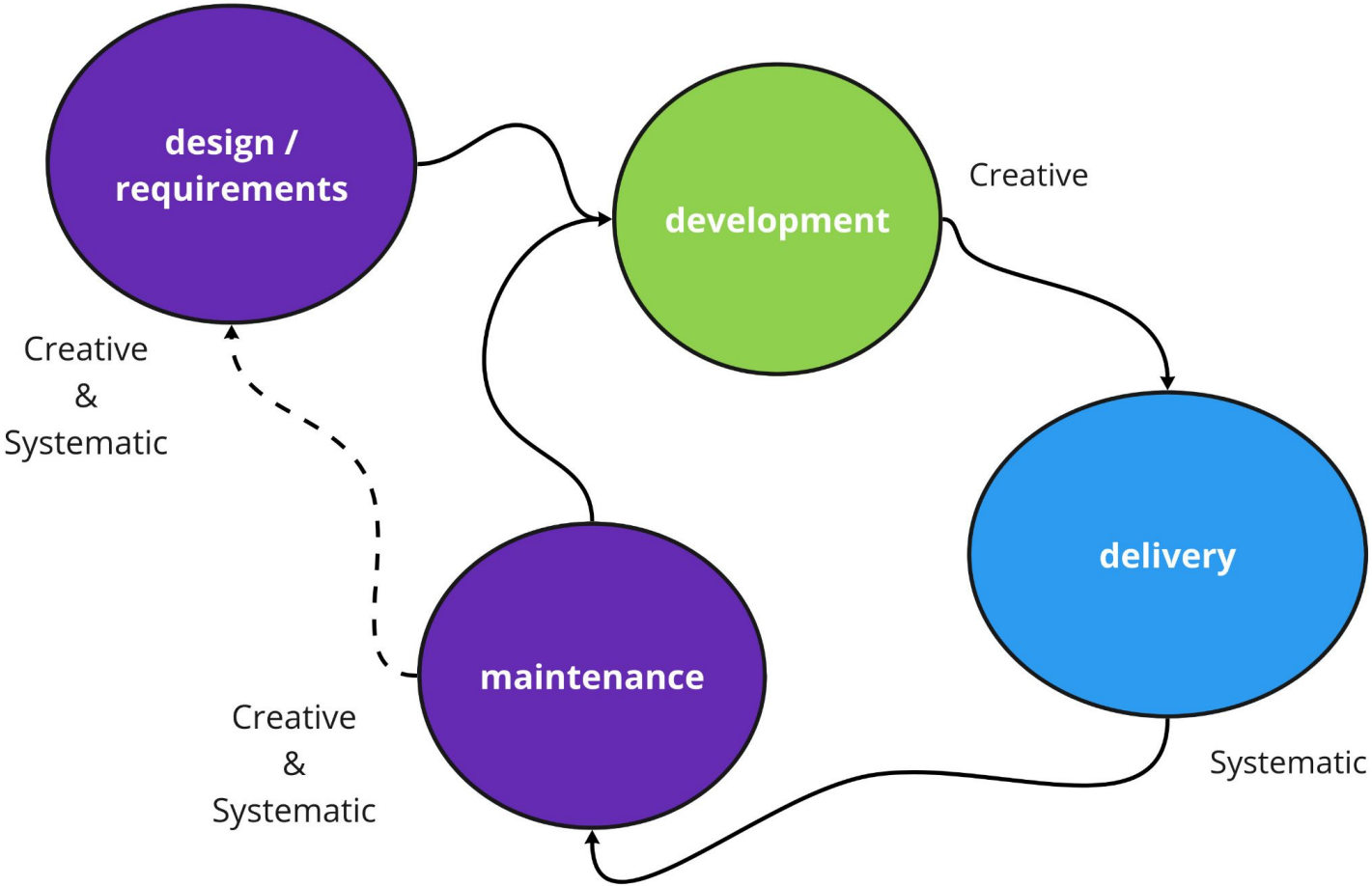
2.- Process using a simple model

2.1.- Modeling our SW product delivery

SW Product Life Cycle (theory)

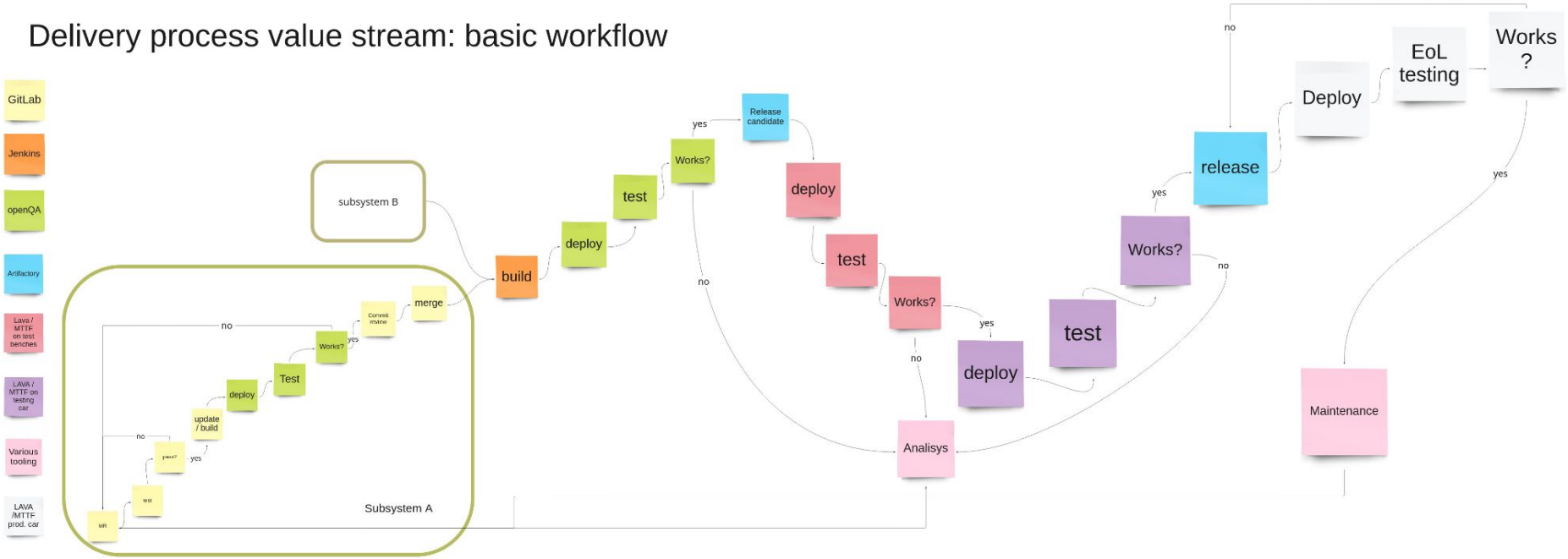


SW Product Life Cycle (reality)



Delivery described: value stream mapping

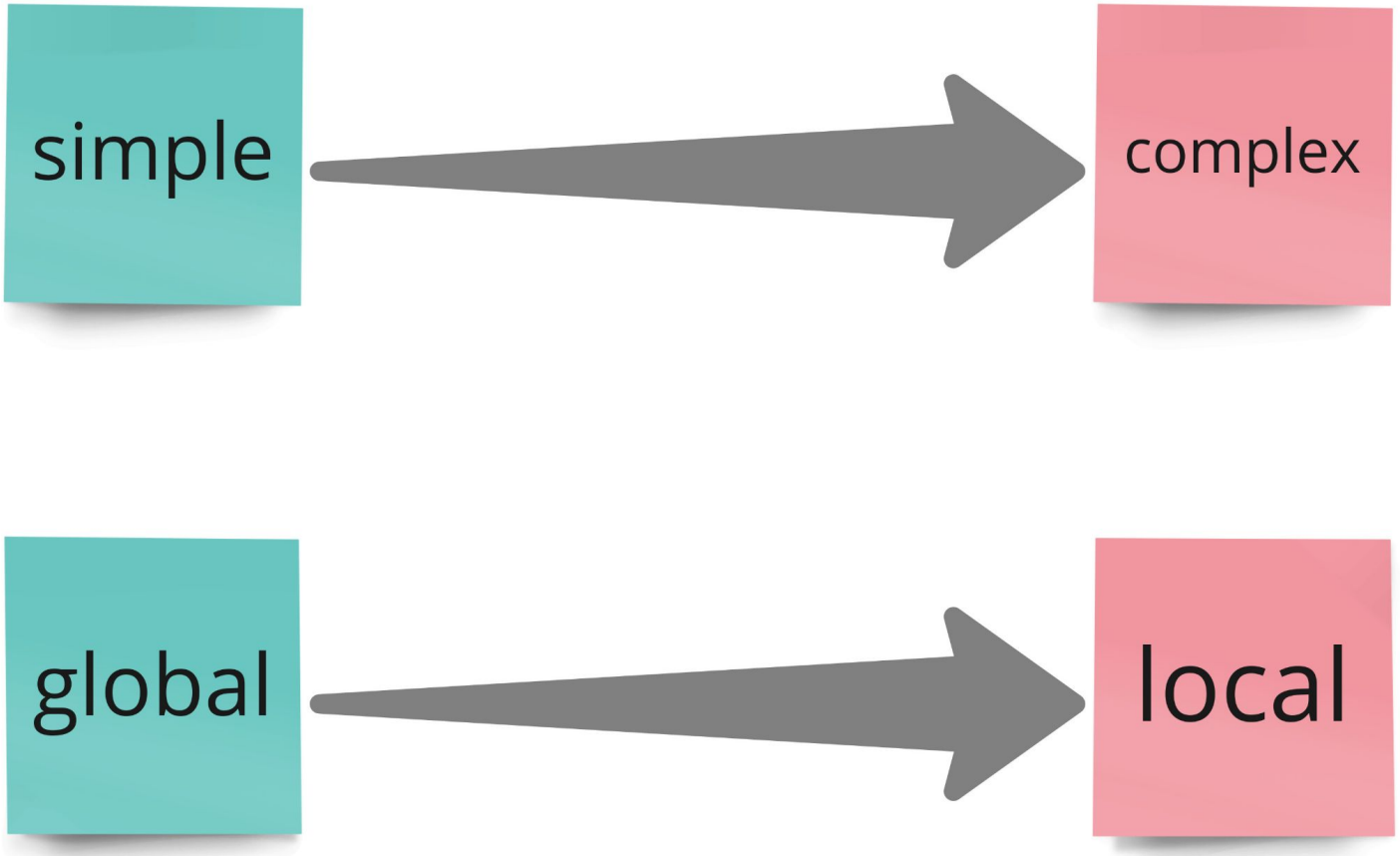
Delivery process value stream: basic workflow



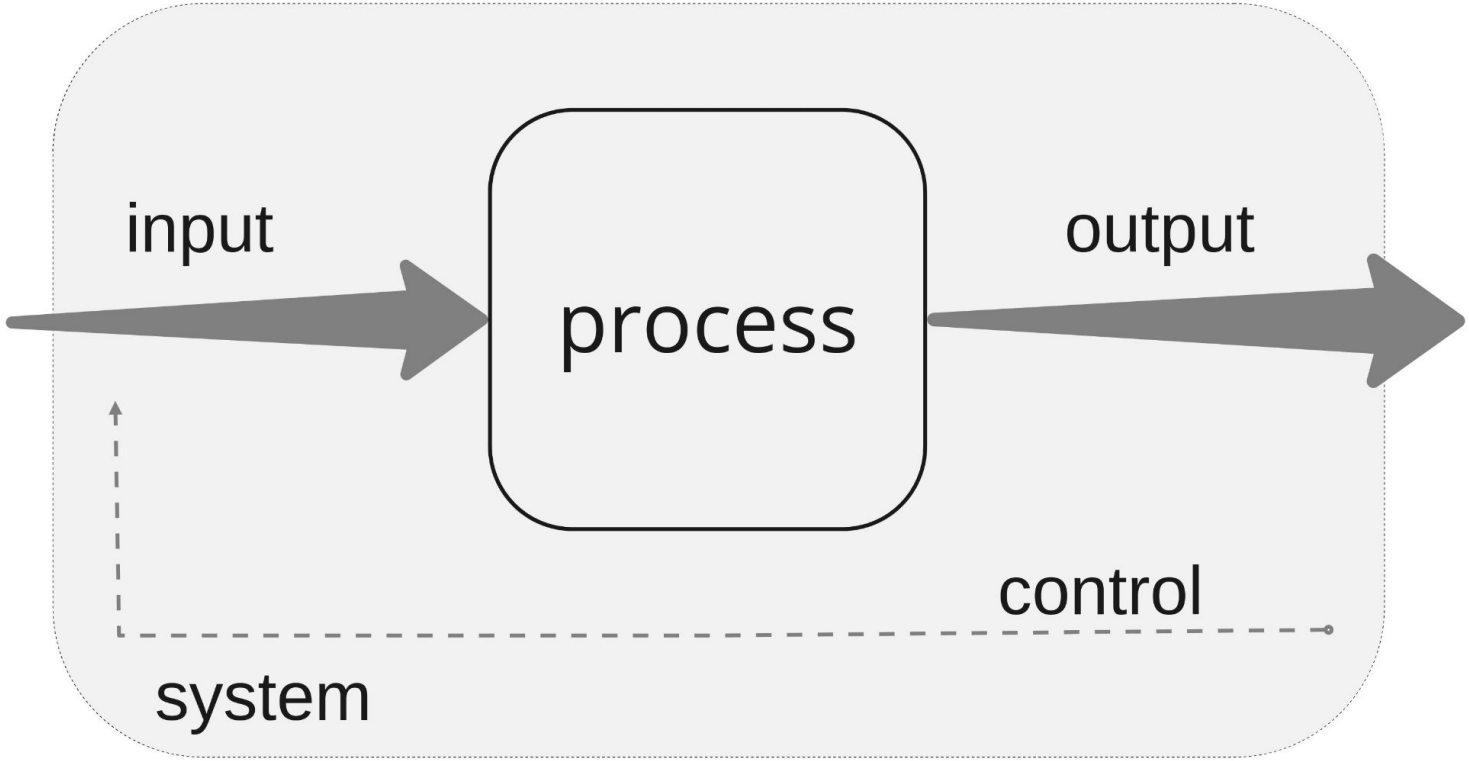
... the sciences do not try to explain, they hardly even try to interpret, they mainly make ***models****. By a model is meant a* ***mathematical construct*** *which, with the addition of certain verbal interpretations,* ***describes observed phenomena****. The justification of such a mathematical construct is solely and precisely that* ***it is expected to work****—that is, correctly to describe phenomena from a reasonably wide area.*- John von Neumann.

Systems thinking is a holistic approach to analysis that focuses on the way that a system's constituent parts interrelate and how systems work over time and within the context of larger systems.

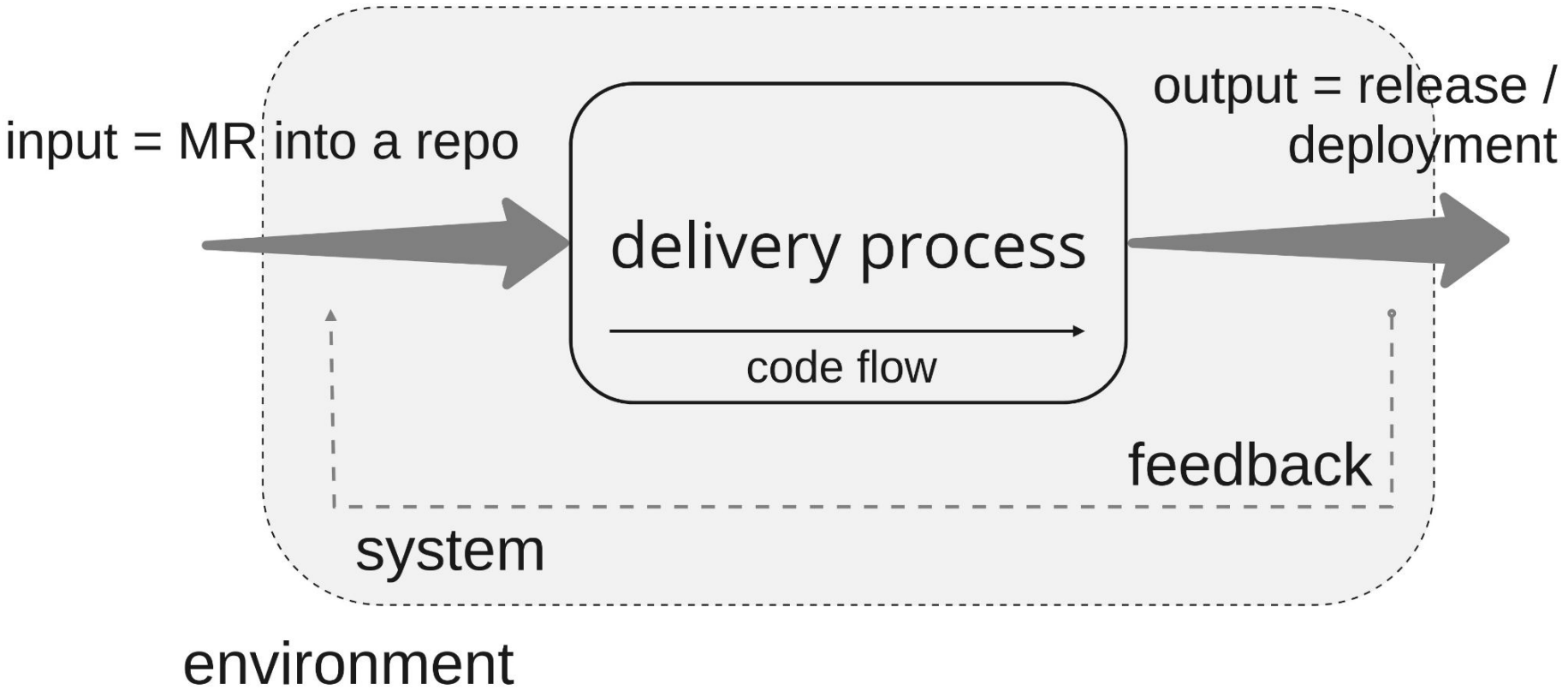
Modelling applying system thinking



The simplest model of a system



The simplest SW product delivery model



The simplest SW product delivery model

We are interested in understanding:

- How well the delivery process performs (**performance**) in our case, that is, how efficient/productive is.
- Some translation to the lean concept of delivery accuracy, which in case of software we can associate to **quality**.

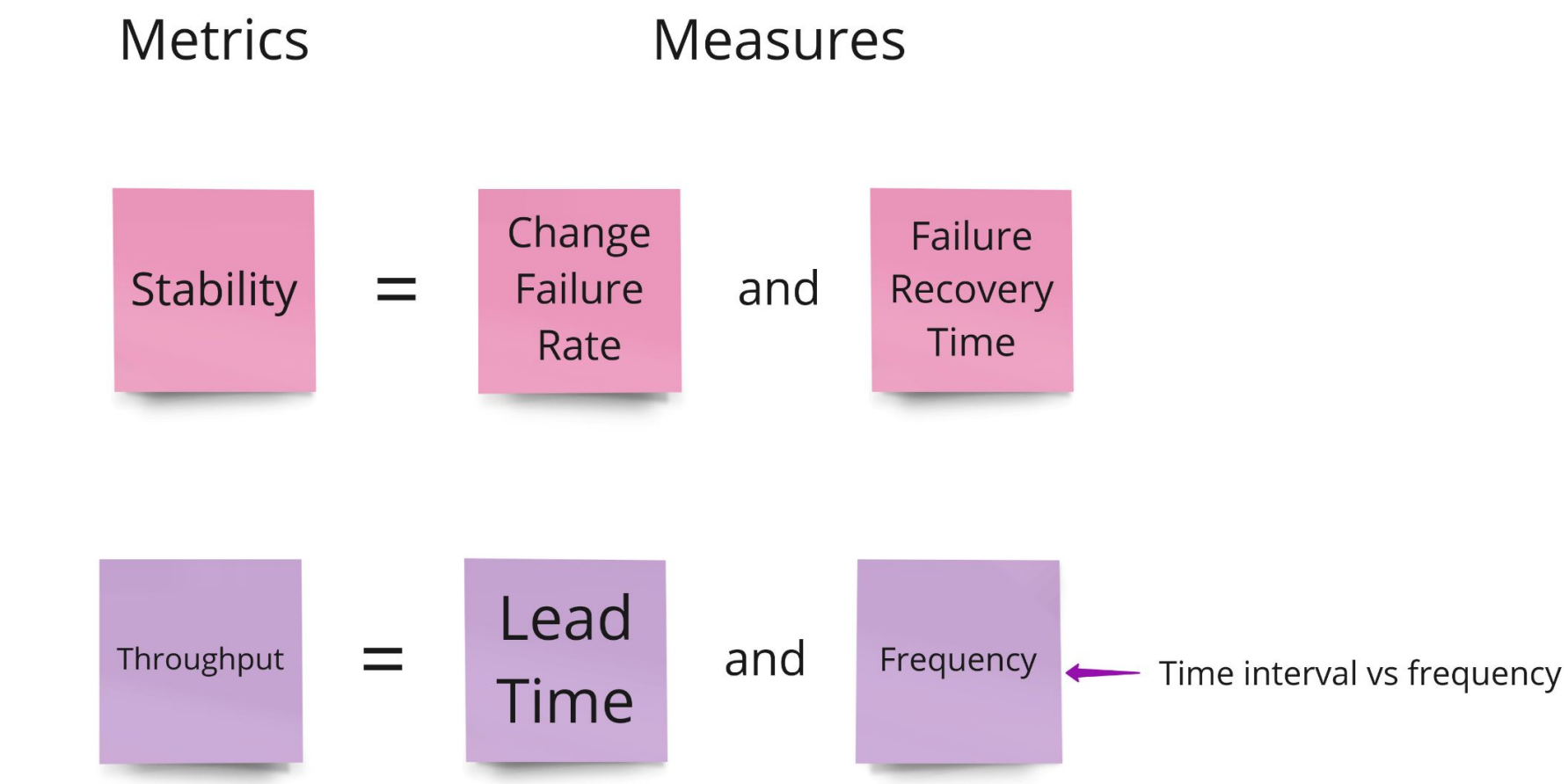
In general, we want to understand the **value** we are delivering to the customer, not only linked to the delivery process but to the entire product life cycle.

2.2.– Define the mathematical construct: metrics characterization

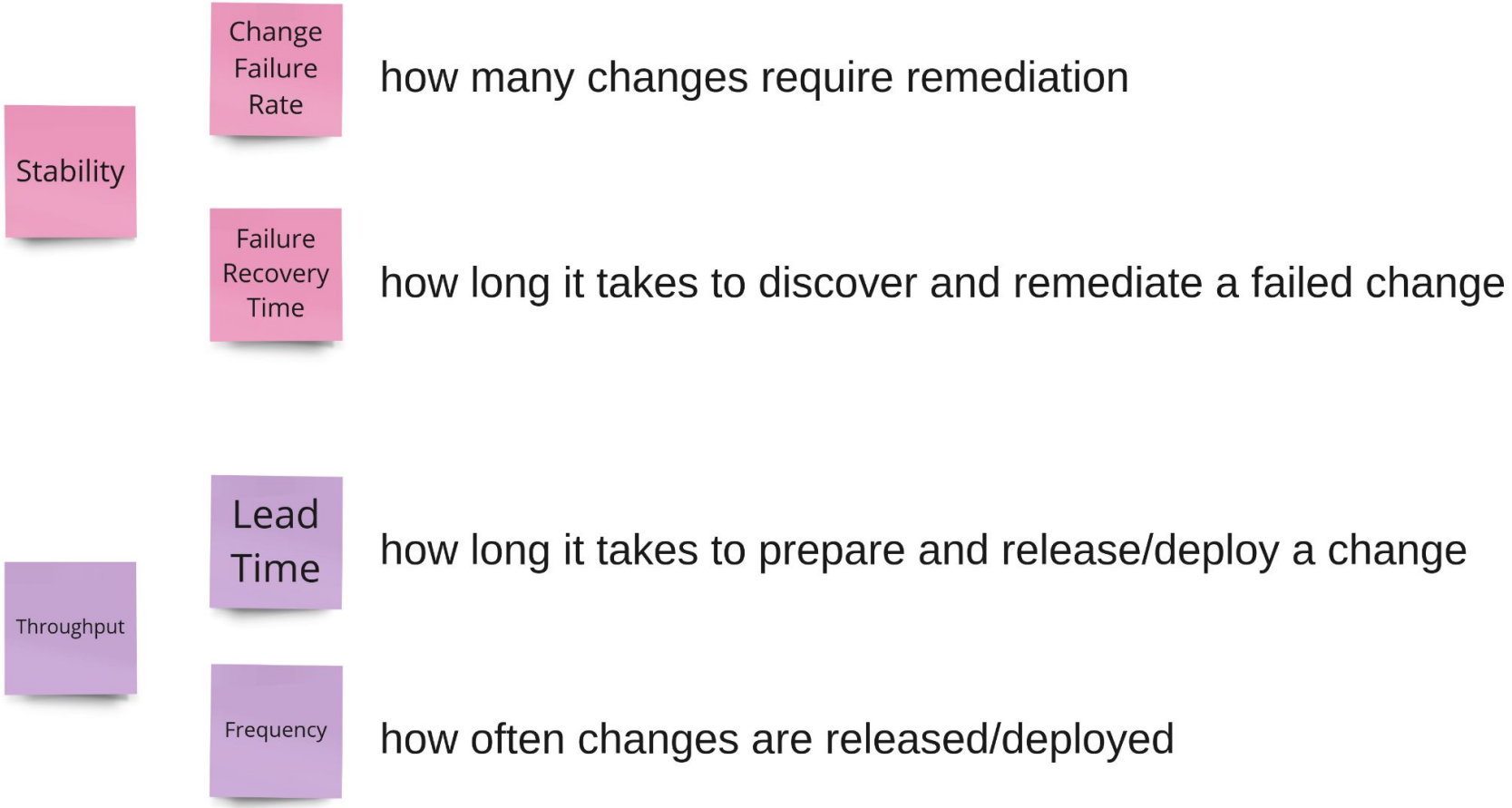
Delivery metrics: description

- **Stability** is a measure of how reliable and robust is our process which has a close relation with the **quality** of the output.
- **Throughput** measures the efficiency with which our delivery process produces the output. It has a close relation with the **performance** of the process.
- **Cost of delay** combines an understanding of **value** with how that value leaks away over time.

Delivery metrics: measures



Delivery metrics: measures definition



Delivery metrics: measures characterization

Stability = Change Failure Rate % Failure Recovery Time seg., min., h, d

Throughput = Lead Time seg., min., h, d Time interval seg., min., h, d

Data sets characterization

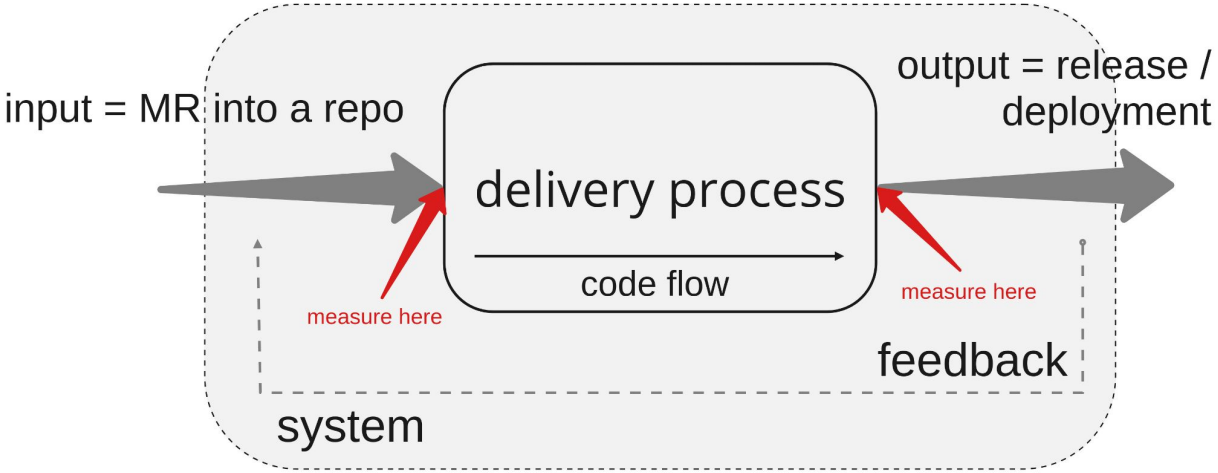
Data set characterised by the average and standard deviation, also referred as variation.

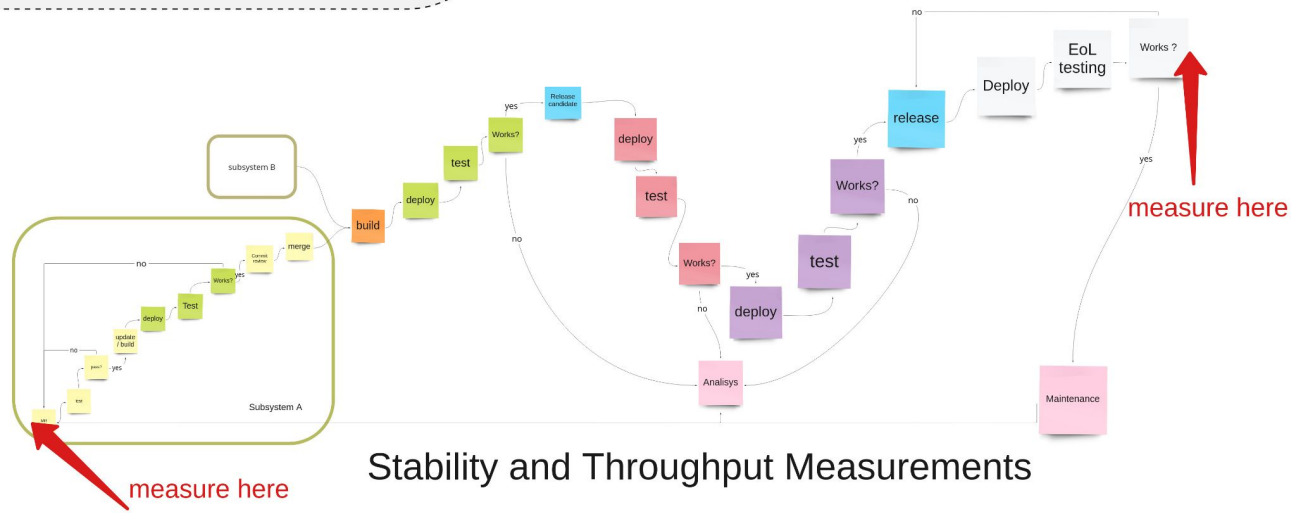
Remember…

If you only quantify one thing, quantify Cost of Delay. – Donald G. Reinertsen

2.3.- Measure, Plot and Quantitative Analysis

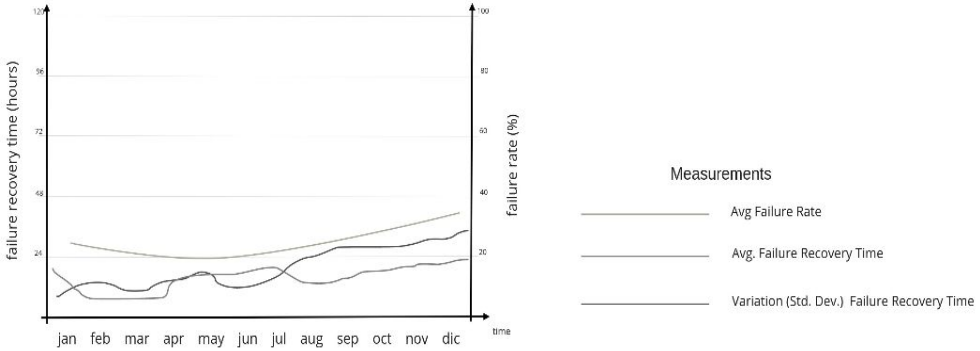
Measure


input = MR into a repo
delivery process
code flow
output = release / deployment
measure here
measure here
feedback
system
environment

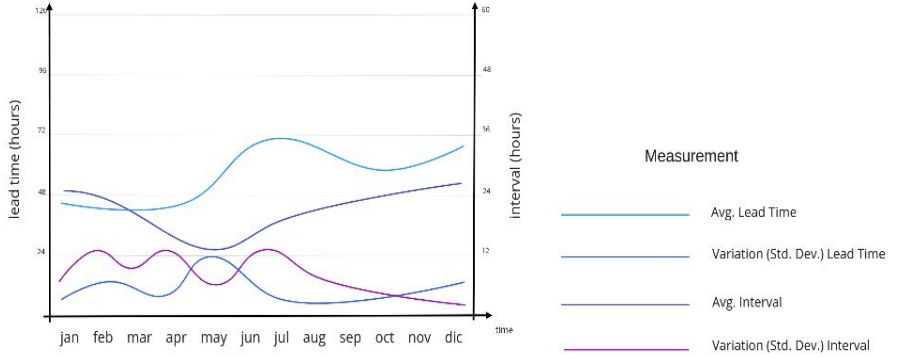

subsystem B
build
deploy
test
Works?
Release candidate
deploy
test
Works?
release
Deploy
EoL testing
Works ?
Works?
test
deploy
Analisys
Maintenance
Subsystem A
merge
Test
deploy
measure here
measure here
Stability and Throughput Measurements

Plot

Stability



Throughput



2.4.- From Quantitative to Qualitative Analysis

Define thresholds

Stability = Change Failure Rate

0.1

	BELOW	ABOVE
Avg	<0.1	>0.1

Failure Recovery Time

	BELOW	ABOVE
Avg		

	BELOW	ABOVE
Std. Dev		

Throughput = Lead Time

	BELOW	ABOVE
Avg		

	BELOW	ABOVE
Std. Dev		

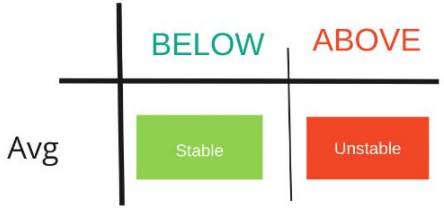
Time interval

	BELOW	ABOVE
Avg		

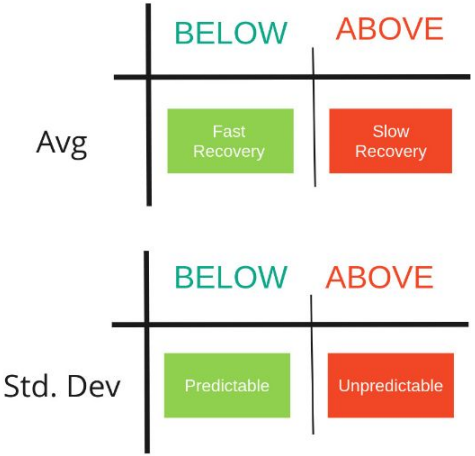
	BELOW	ABOVE
Std. Dev		

Define trends

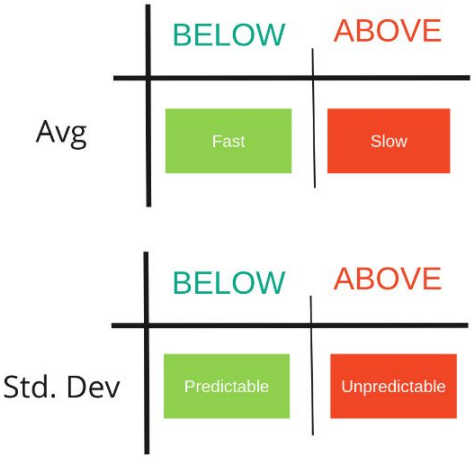
Stability = Change Failure Rate



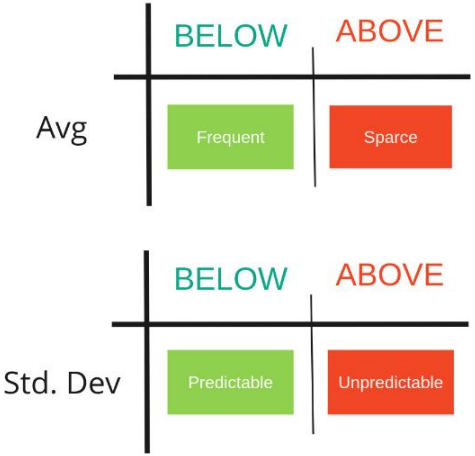
Failure Recovery Time



Throughput = Lead Time



Time interval



Define scenarios

Throughput Scenarios

Metric	Trend / Measures	Slow & infrequent	Slow & unpredictable	Unpredictable & infrequent	Fast & Frequent	Too Fast & Too Frequent
Lead Time	Average Lead Time	ABOVE	BELOW	BELOW	BELOW	BELOW
	Variation in Lead Time	ABOVE	ABOVE	N/A	BELOW	BELOW
Time Interval	Average Time Interval	ABOVE	N/A	BELOW	BELOW	BELOW
	Variation in Time Interval	ABOVE	N/A	ABOVE	BELOW	BELOW

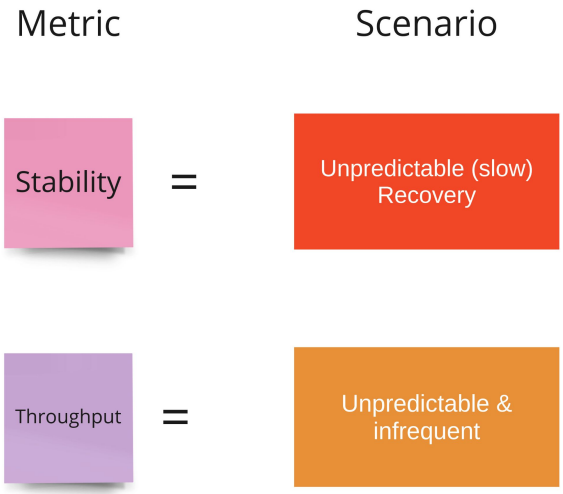
Define scenarios

Stability Scenarios

Metric	Trend / Measures	Unstable	Slow Recovery	Unpredictable (slow) Recovery	Fast Recovery	Stability
Failure Rate	Average Failure Rate	ABOVE	N/A	N/A	ABOVE	BELOW
Failure Recovery Time	Average Failure Recovery Time	ABOVE	ABOVE	BELOW	BELOW	BELOW
	Variation in Failure Recovery Time	ABOVE	BELOW	ABOVE	BELOW	BELOW

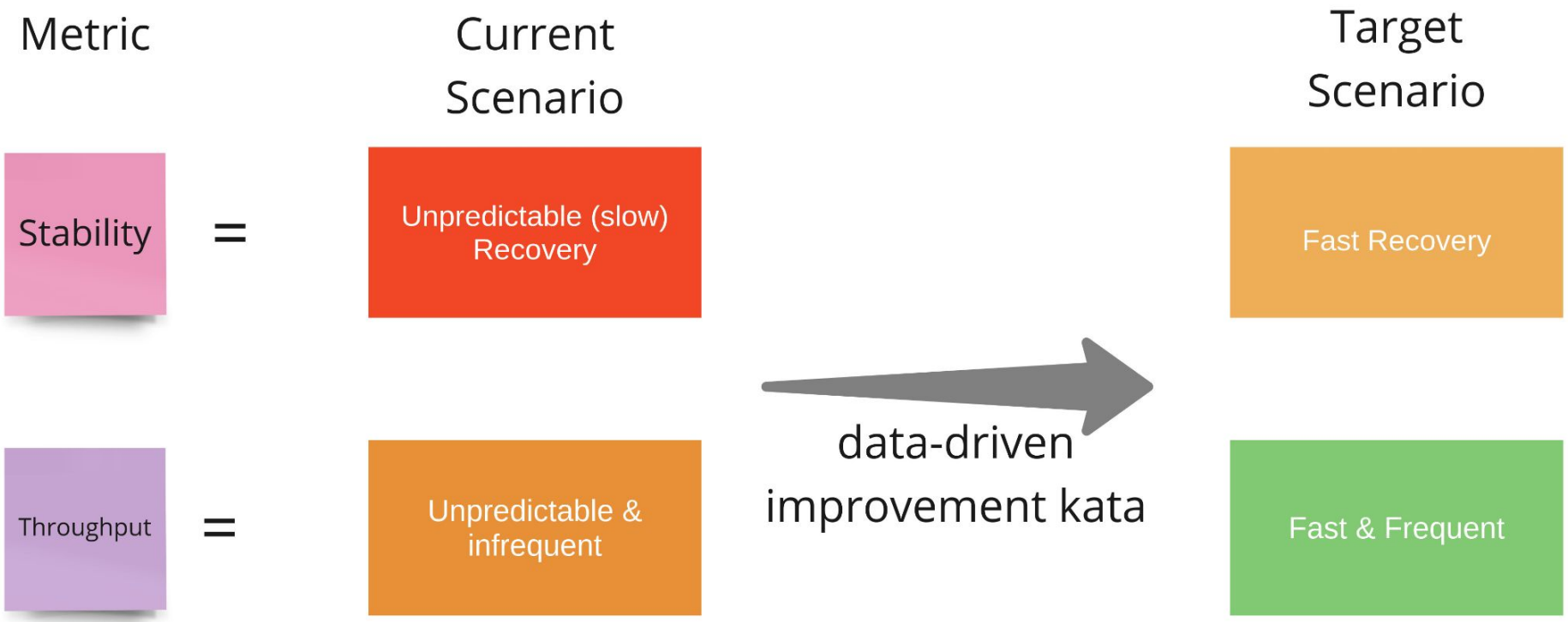
SW product delivery metrics: summary and example

ID	Metric	Description	Measures	Measurement process	Associated indicators	Threshold	Scenarios
CoD	Cost of Delay						
T	Throughput						
S	Stability						



2.5.- From our current scenario to a target one: Continuous Improvement

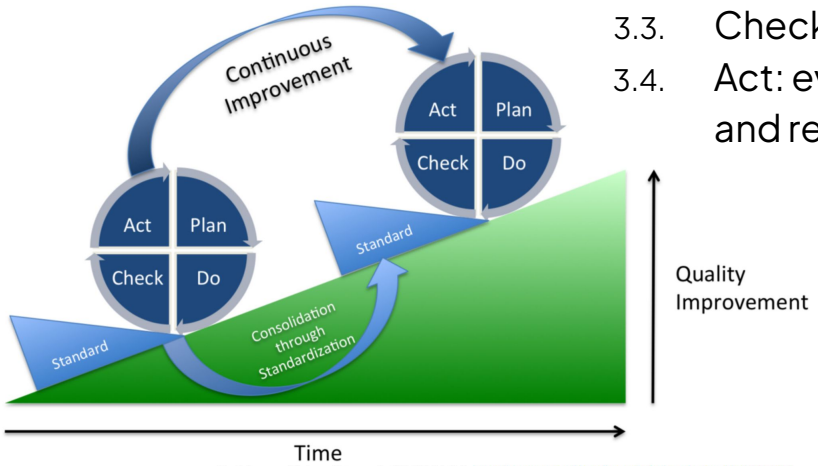
From current to target scenario: example



Improvement kata definition

Key steps:

1. Define success (business/product/technical): goal
2. Plan the improvement cycle.
3. Execute: PDCA
   - 3.1. Plan: define the experiments
   - 3.2. Do: execute the experiments
   - 3.3. Check: evaluate the experiments
   - 3.4. Act: evaluate, consolidate the experiments when appropriate and re-define the target conditions.



Data-driven improvement kata

The continuous improvement process to follow is not just any kind of improvement kata, but a **data-driven** one. Every PDCA step is defined, tracked, evaluated and re-defined **based on the delivery metrics**.

Data-driven improvement kata: board

Data Driven Improvement Kata Board

Priority	Action	Description	Current Condition	Target condition	Goal
1	Action / experiment name	Short description of the experiment: hypothesis, experiment, analysis and conclusions	• Stability: • Throughput: • Other:	• Stability: • Throughput: • Other	• Stability: • Throughput: • Scenario:
2					
3					

- Each business goal refers to a cycle. It is recommended to define 1 year as initial cycle.
  - Each goal should be defined based on Stability and Throughput metrics. For instance: to achieve X
    - Stability will increase by 50%, so failure change rate should be lower than 5% and Failure recovery time no longer than 5 hours.
    - Throughput will increase by 25% so lead time will go down to 12 hours and time interval down to 2 hours.
- The values on the "Current Condition" column come from the existing data.
- The target condition correspond to the goal for the coming iteration. It should also be defined in terms of overall stability and throughput.
- Limit the number of goals to 3 or 4, including at least one described in terms of a financial-type of value that can be related to CoD.

3.– Iterating the process. Addressing complexity

Define a more complex model

- This process works for a simple model, which provides highly useful insides.
- If it is the right process, it should work for more complex models too.
- What about the key elements of each step of the process?

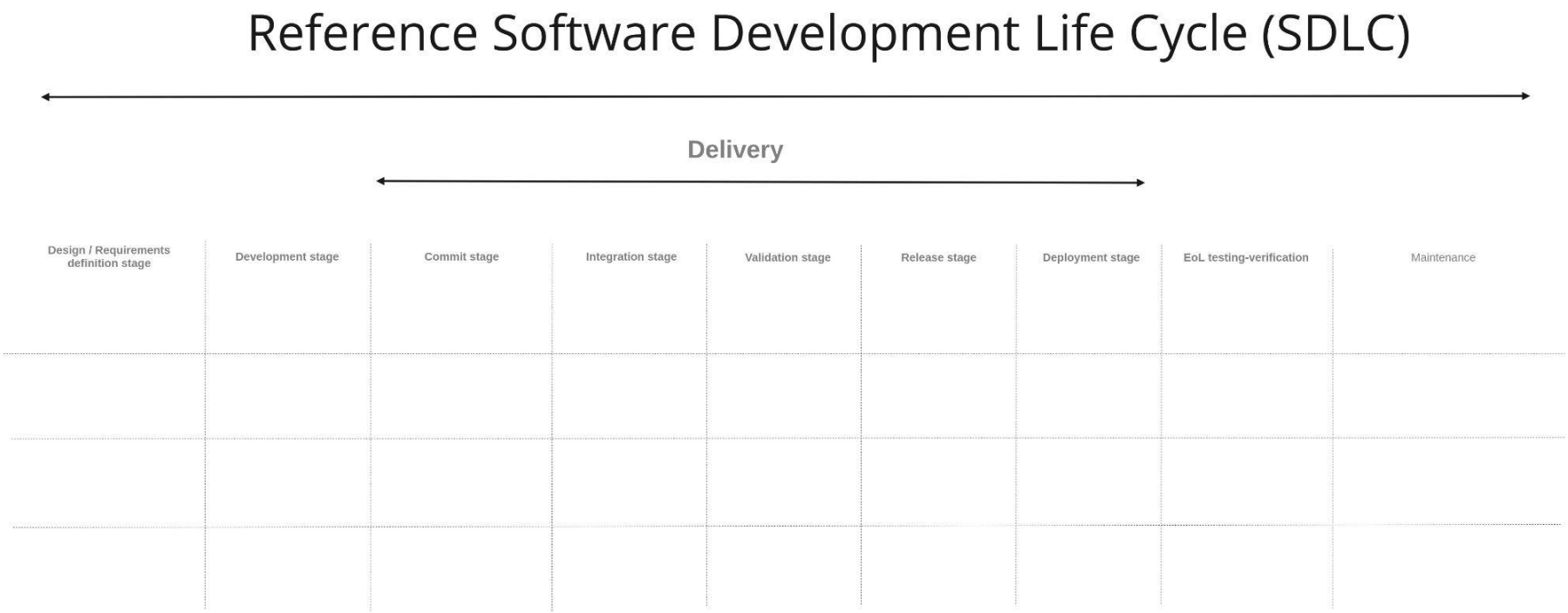
The process when creating complex models

As expected, the procedure is basically the same:

1. Structure your delivery process and create a model
2. Characterise the metrics to measure performance
3. Measure, plot and perform a quantitative analysis of the model
4. Move towards a qualitative analysis
5. Data-driven improvement kata to increase performance moving from the current to the target scenario.

3.1.- Creating a more complex model

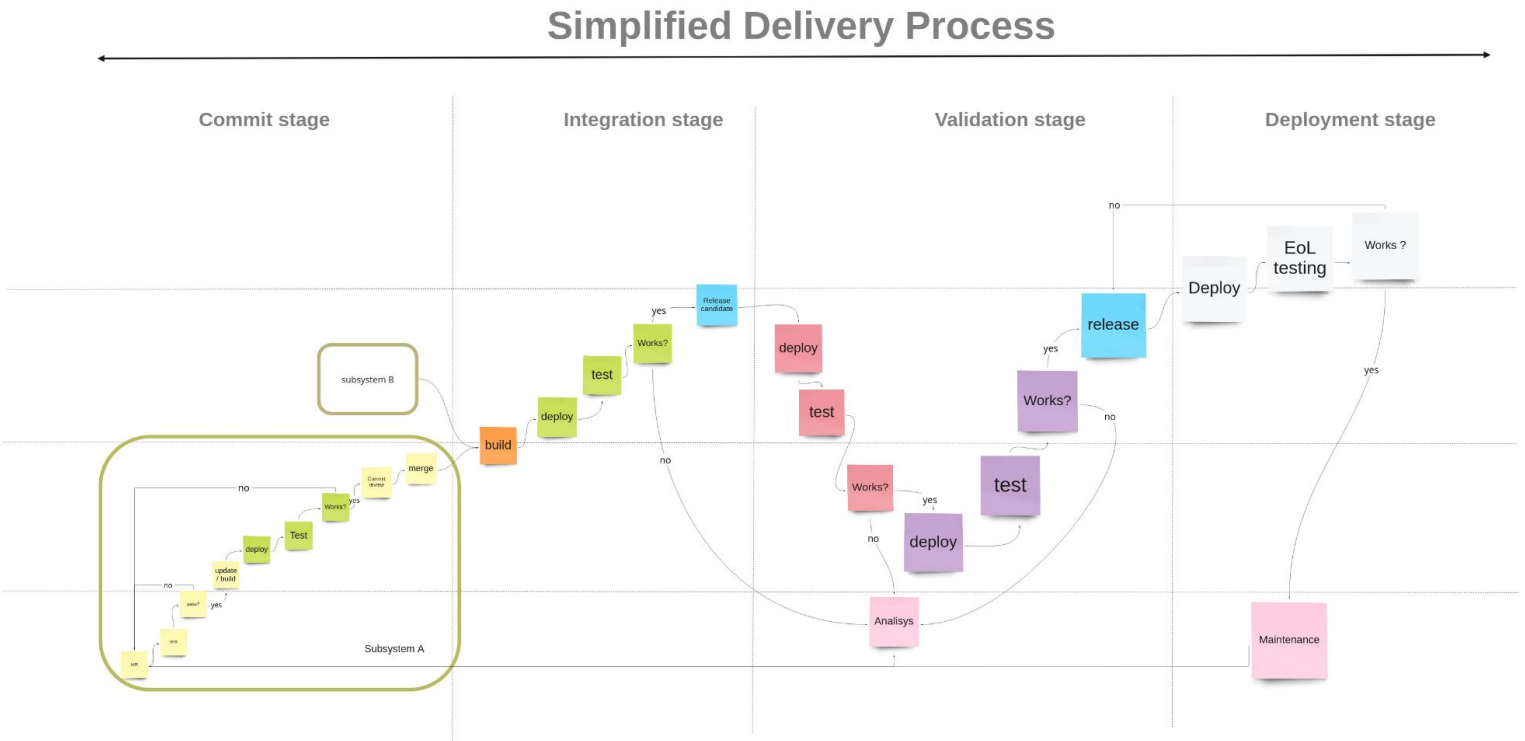
Definition of the product life cycle at high level



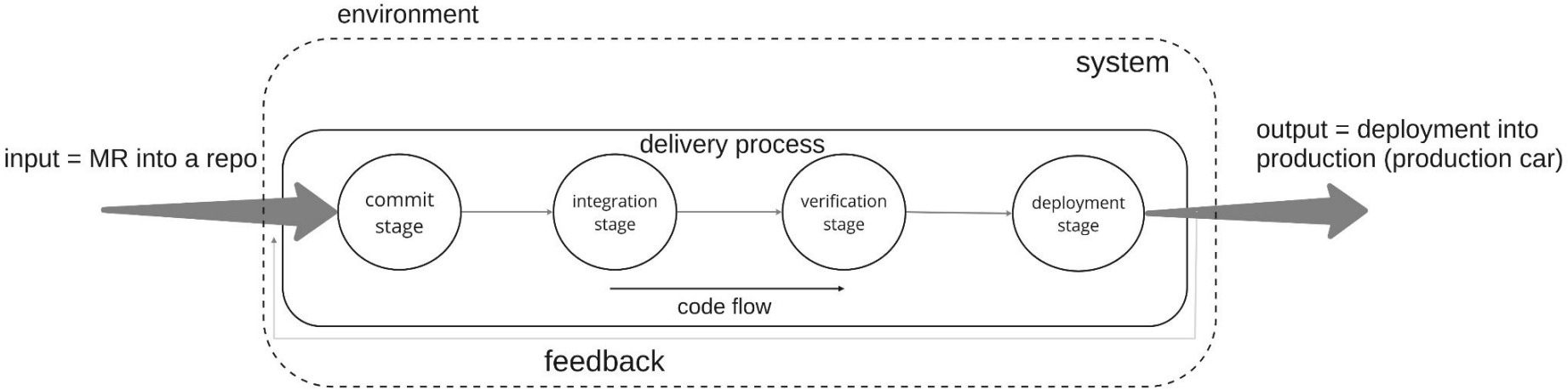
Definition of the delivery process at high level

Structure your delivery process

Modelling your delivery process



New SW product delivery model



3.2.- Mathematical construct: metrics characterization for our new model

Metrics for the new model: definition

Metric ID	Name	Description	Measurement process	Associated Indicators	Trends and target threshold	Scenarios
CoD	Cost of Delay					
T	Throughput					
S	Stability					

Same metrics!

Metrics Measures

Stability = Change Failure Rate and Failure Recovery Time

Throughput = Lead Time and Frequency ← Time interval vs frequency

Indicators for each (sub)system of the model

Indicators definition

Simplified Delivery Process

Commit stage	Integration stage	Validation stage	Deployment stage
MTI Master Throughput Indicator	ITI Build (integration) Throughput Indicator	VTI Validation Throughput Indicator	DTI Deployment Throughput Indicator
	ISI Build (integration) Stability Indicator	VSI Validation Stability Indicator	DSI Deployment Stability Indicator

Commit stage

Indicator		Measures		
Master Throughput Indicator	=	Master Lead Time	and	Master Interval

Integration (build) stage

Indicator		Measures		
Integration Stability Indicator	=	Integration Change Failure Rate	and	Integration Failure Recovery Time
Integration Throughput Indicator	=	Integration Lead Time	and	Integration Interval

Validation stage

Indicator		Measures		
Validation Stability Indicator	=	Validation Change Failure Rate	and	Validation Failure Recovery Time
Validation Throughput Indicator	=	Validation Lead Time	and	Validation Interval

Deployment stage

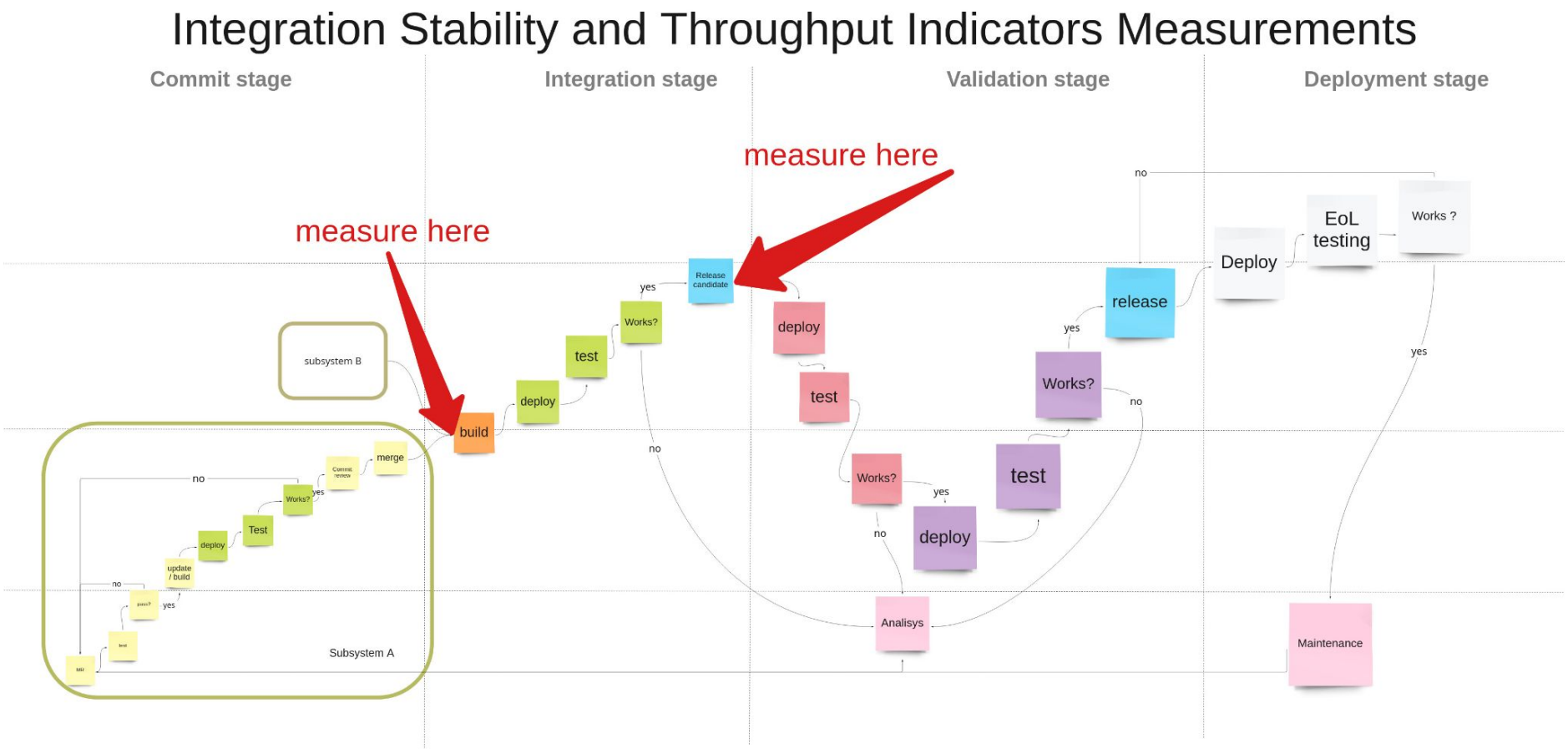
Indicator		Measures		
Deployment Stability Indicator	=	Deployment Change Failure Rate	and	Deployment Failure Recovery Time
Deployment Throughput Indicator	=	Deployment Lead Time	and	Deployment Interval

Same metrics

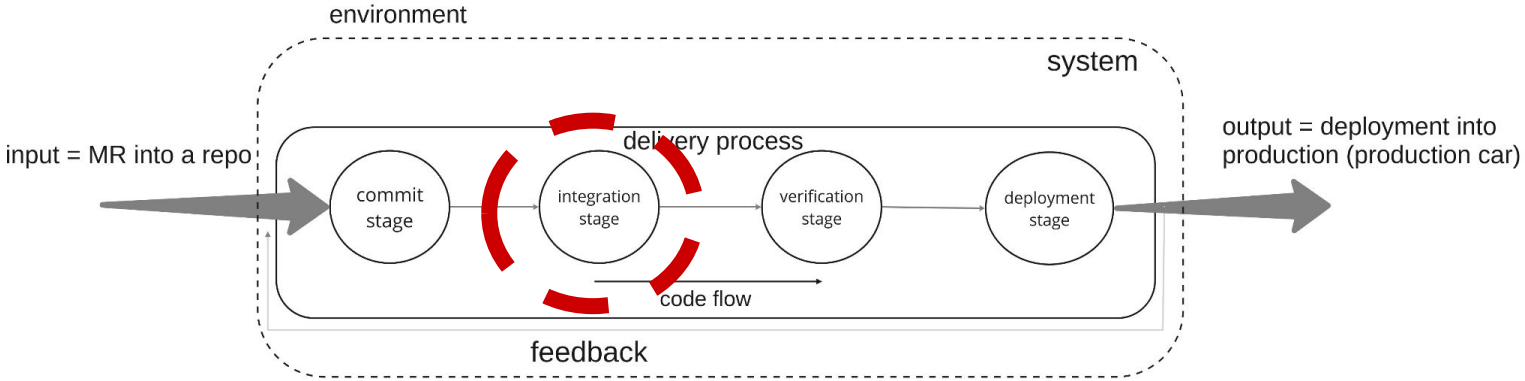
- Metrics for the simplified and this more detailed model are the same.
- Those metrics are consistently applied to each stage and the overall system.
- We can measure the impact of a local experiment locally, on each of the other (sub)systems and the overall system.

3.3.– Measure, Plot and Quantitative Analysis

Measure each indicator

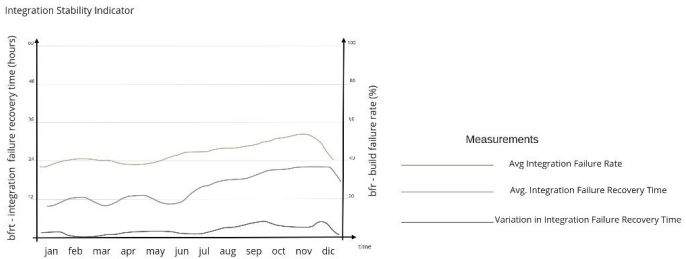


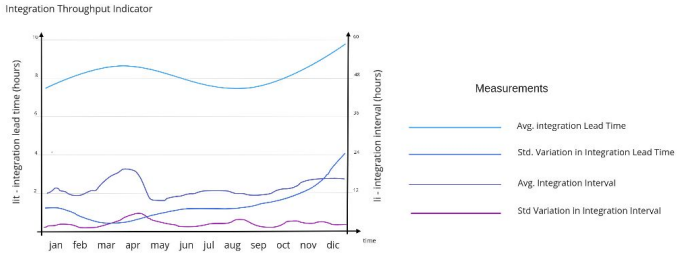
Measure each indicator



Integration (build) stage

Indicator		Measures		
Integration Stability Indicator	=	Integration Change Failure Rate	and	Integration Failure Recovery Time
Integration Throughput Indicator	=	Integration Lead Time	and	Integration Interval

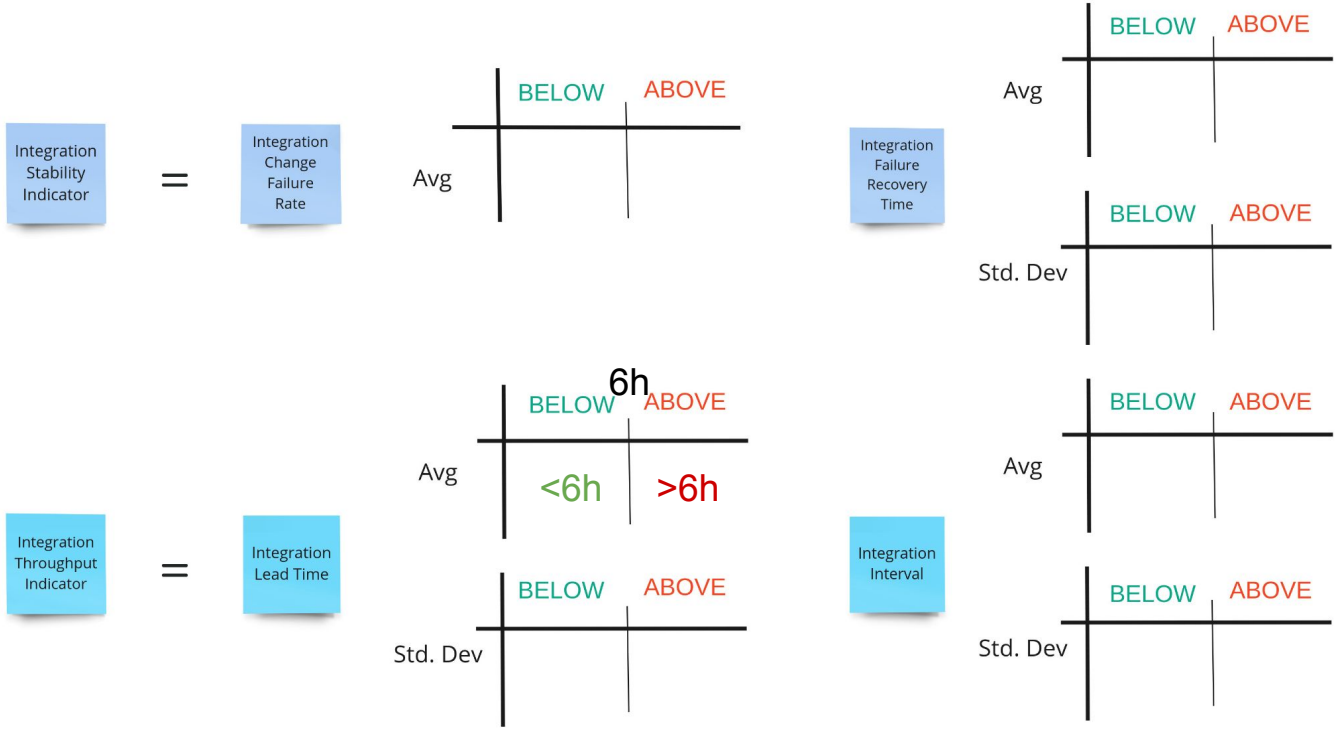




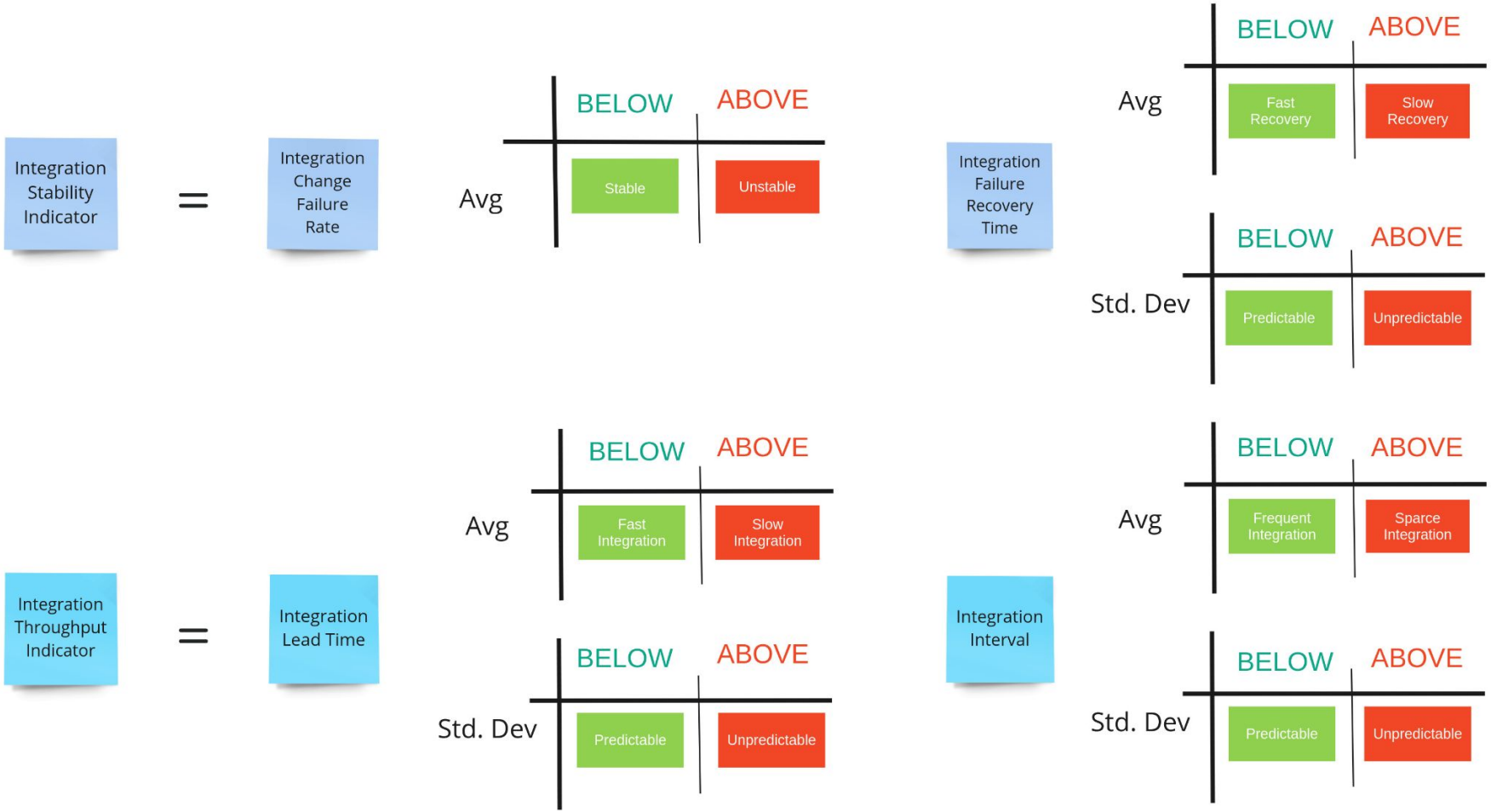
3.4.- From Quantitative to Qualitative Analysis

Define thresholds for each indicator: example

Now, in addition to a threshold for each of the measures corresponding to both metrics applied to the model of the entire system, you need to define thresholds for each measure of each indicator (persubsystem/stage).



Define trends for each indicator: example



Define scenarios

Integration Stability Indicator

Trend \ Metrics	Integration/build Failure Rate: Average Integration failure rate	Integration/build Failure Recovery time: Average Integration Failure Recovery Time	Integration/build Failure Recovery time: Variation in Integration Failure Recovery Time
Unstable	ABOVE	ABOVE	ABOVE
Slow Recovery	N/A	ABOVE	BELOW
Unpredictable (slow) Recovery	N/A	BELOW	ABOVE
Fast Recovery	ABOVE	BELOW	BELOW
Stability	BELOW	BELOW	BELOW

Integration Throughput Indicator

Trend \ Metrics	Integration/build Lead Time: Average Integration Lead Time	Integration/build Lead Time: Variation in Integration Lead Time	Integration/build Interval: Average Integration Interval	Integration/build Interval: Variation in Integration Interval
Slow & infrequent builds	ABOVE	ABOVE	ABOVE	ABOVE
Slow & unpredictable builds	BELOW	ABOVE	N/A	N/A
Fast & infrequent builds	BELOW	BELOW	ABOVE	N/A
Unpredictable & infrequent builds	BELOW	N/A	BELOW	ABOVE
Too fast & too frequent builds	BELOW	BELOW	BELOW	BELOW
Fast & frequent builds	BELOW	BELOW	BELOW	BELOW

Commit stage

Indicator | Measures

Master Throughput Indicator = Master Lead Time and Master Interval

Integration (build) stage

Indicator | Measures

Integration Stability Indicator = Integration Change Failure Rate and Integration Failure Recovery Time

Integration Throughput Indicator = Integration Lead Time and Integration Interval

Master Throughput Indicator

Metrics / Trend	Master Lead Time		Master Interval	
	Average Master Lead Time	Variation in Master Lead Time	Average Master Interval	Variation in Master Interval
Slow & infrequent feedback to Master	HIGH	HIGH	HIGH	HIGH
Slow & unpredictable commits to Master	LOW	HIGH	N/A	N/A
Unpredictable & infrequent commits to Master	LOW	N/A	LOW	HIGH
Too fast & too frequent commits to Master	LOW	LOW	LOW	LOW
Fast & frequent commits to Master	LOW	LOW	LOW	LOW

Build Stability Indicator

Metrics / Trend	Integration/build Failure Rate	Integration/build Failure Recovery time	
	Average Build failure rate	Average Build Failure Recovery Time	Variation in Build Failure Recovery Time
Unstable	HIGH	HIGH	HIGH
Slow Recovery	N/A	HIGH	LOW
Unpredictable (slow) Recovery	N/A	LOW	HIGH
Fast Recovery	HIGH	LOW	LOW
Stability	LOW	LOW	LOW

Build Throughput Indicator

Metrics / Trend	Integration/build Lead Time		Integration/build Interval	
	Average Build Lead Time	Variation in Build Lead Time	Average Build Interval	Variation in Build Interval
Slow & infrequent builds	HIGH	HIGH	HIGH	HIGH
Slow & unpredictable builds	LOW	HIGH	N/A	N/A
Fast & infrequent builds	LOW	LOW	HIGH	N/A
unpredictable & infrequent builds	LOW	N/A	LOW	HIGH
Too fast & too frequent builds	LOW	LOW	LOW	LOW
Fast & frequent builds	LOW	LOW	LOW	LOW

Validation stage

Indicator | Measures

Validation Stability Indicator = Validation Change Failure Rate and Validation Failure Recovery Time

Validation Throughput Indicator = Validation Lead Time and Validation Interval

Deployment stage

Indicator | Measures

Deployment Stability Indicator = Deployment Change Failure Rate and Deployment Failure Recovery Time

Deployment Throughput Indicator = Deployment Lead Time and Deployment Interval

Validation Stability Indicator

Metrics / Trend	Validation Failure Rate	Validation Failure Recovery time	
	Average Validation failure rate	Average Validation Failure Recovery Time	Variation in Deployment Failure Recovery Time
Inevitability	HIGH	HIGH	HIGH
Slow Recovery	N/A	HIGH	LOW
Unpredictable Recovery	N/A	LOW	HIGH
Fast Recovery	HIGH	LOW	LOW
Stability	LOW	LOW	LOW

Validation Throughput Indicator

Metrics / Trend	Validation Lead Time		Validation Interval	
	Average Deployment Lead Time	Variation in Deployment Lead Time	Average Deployment Interval	Variation in Deployment Interval
Slow & infrequent	HIGH	HIGH	HIGH	HIGH
Slow & unpredictable	LOW	HIGH	HIGH	N/A
Fast & infrequent	LOW	LOW	HIGH	N/A
unpredictable & infrequent	LOW	N/A	LOW	HIGH
Too fast & too frequent	LOW	LOW	LOW	LOW
Fast & frequent	LOW	LOW	LOW	LOW

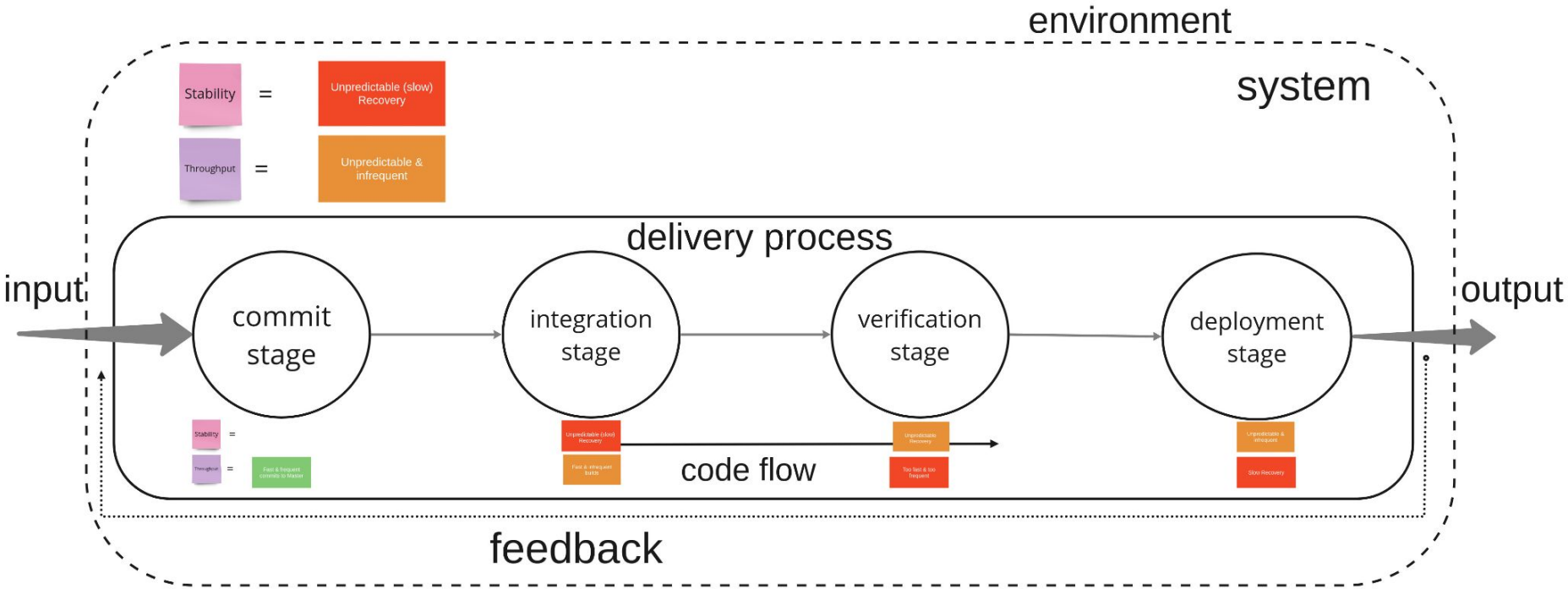
Deployment Stability Indicator

Metrics / Trend	Deployment Failure Rate	Deployment Failure Recovery time	
	Average Deployment failure rate	Average Deployment Failure Recovery Time	Variation in Deployment Failure Recovery Time
Inevitability	HIGH	HIGH	HIGH
Slow Recovery	N/A	HIGH	LOW
Unpredictable Recovery	N/A	LOW	HIGH
Fast Recovery	HIGH	LOW	LOW
Stability	LOW	LOW	LOW

Deployment Throughput Indicator

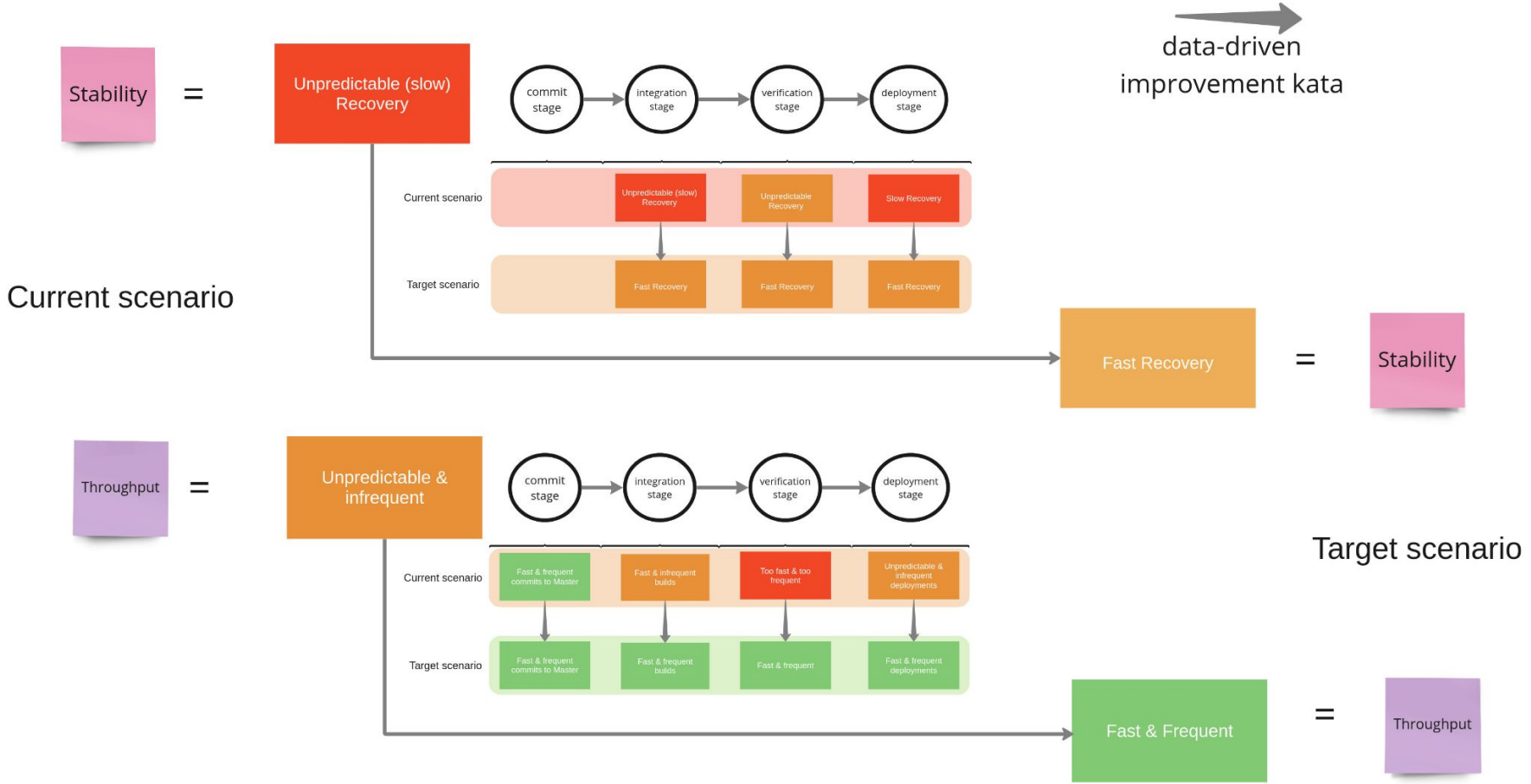
Metrics / Trend	Deployment Lead Time		Deployment Interval	
	Average Deployment Lead Time	Variation in Deployment Lead Time	Average Deployment Interval	Variation in Deployment Interval
Slow & infrequent	HIGH	HIGH	HIGH	HIGH
Slow & unpredictable	LOW	HIGH	HIGH	N/A
Fast & infrequent	LOW	LOW	HIGH	N/A
unpredictable & infrequent	LOW	N/A	LOW	HIGH
Too fast & too frequent	LOW	LOW	LOW	LOW
Fast & frequent	LOW	LOW	LOW	LOW

Current scenario: example



3.5.- From our current scenario to a target one: Continuous Improvement

From current to target scenario: example



There is no time to cover this…

You can get a short summary of how to approach the data-driven improvement kata for this extended model in the blog post below. In any case, it is highly dependant on the organization structure:

- Improve your software product delivery process performance using metrics (II)

There is plenty of literature on this topic you can check.

4.- Summary

The price of light is less than the cost of darkness. - Arthur C. Nielsen.

Errors using data are much less than those using no data at all. – Charles Babbage

I never guess. It is a capital mistake to theorize before one has data. Insensibly one begins to twist facts to suit theories, instead of theories to suit facts.- Sir Arthur Conan Doyle, Author of Sherlock Holmes stories

The more uncertainty you have, the bigger uncertainty reduction you get from just a few data points. - Douglas W. Hubbard

Goal: turn data…

… into information then …..

… into insights.

The goal is to turn data into information, and information into insight. - Carly Fiorina.

Summary

1. Convinced on the relevance of metrics applied to your delivery system? If not, iterate. If yes, keep reading.
2. Model your delivery system.
3. Select CoD, Throughput and Stability as core delivery process performance metrics (delivery metrics).
4. Measure, plot and analyse the data (quantitative analysis).
5. Move from a quantitative to a qualitative analysis: define thresholds, trends and define scenarios.
6. Which scenario are you in? Where do you want to be? Define a data-driven improvement kata.

5.– More information and references

More information...

This process and the following steps to apply it at scale are described in two blog posts:

- Improve your software product delivery process performance using metrics (I)
- Improve your software product delivery process performance using metrics (II)

Links and References

References.

1. Main reference: “Measuring Continuous Delivery”. Author: Steve Smith Publisher: Leanpub
2. “The Principles of Product Development Flow”. Author: Donald G. Reinerstsen. Publisher: Celebritas Publishing.
3. “Continuous Delivery”.Authors: Jez Humble and David Farley. Publisher: Addison Wesley.
4. There are more references on the Reads section of my site.

Copyright assignments

1. Several of the representations are inspired or based in those included in the book “Measuring Continuous Delivery”, Author: Steve Smith Publisher: Leanpub.
2. Emojis. Creazilla.com Pubic Domain. Slides 67 and 68.
3. Blind man pic. Slide 66
4. Improvement kata. By Johannes Vietze - Own work, CC BY-SA 3.0. Slide 34

Thank You!

- to the OpenSouthCode organisers for inviting me as speaker.
- to you for attending to this dense talk...during the last slot of the event!

Uso de Métricas en Producción de SW. a Escala

Delivery metrics

OpenSouthCode 2023
2023-06-10
Málaga. Spain

Agustín Benito Bethencourt
Independent consultant