

VIESMANN

The role of Quality Goals and real time KPIs monitoring

Hi !!

- Michał Buczko
- Software Quality Engineering Manager
- 18+ years in testing
- 3 years at Viessmann working only focused on KPIs, dashboards and delivery monitoring



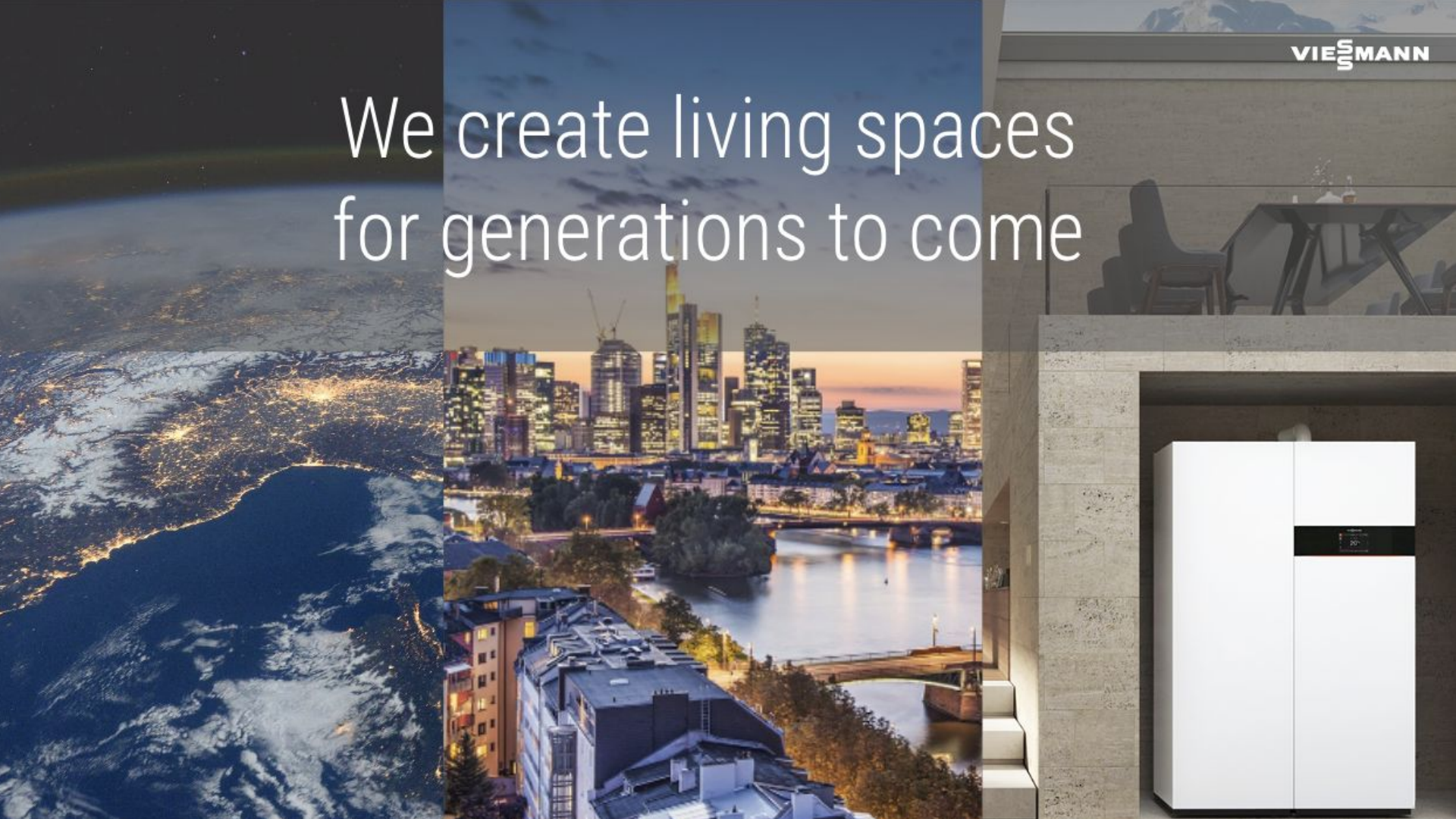
I got inspired,

By Basque Country - Athletic Club,
By the conference.

What is my background?

We create living spaces
for generations to come

VIESSMANN



Climate change is
the greatest
challenge of our
times ...

... demanding every
business to become a
1.5° company.



1t less CO₂/year
for each
modernized boiler

Reach potential:
7 million outdated
non-condensing
boilers in Germany
alone

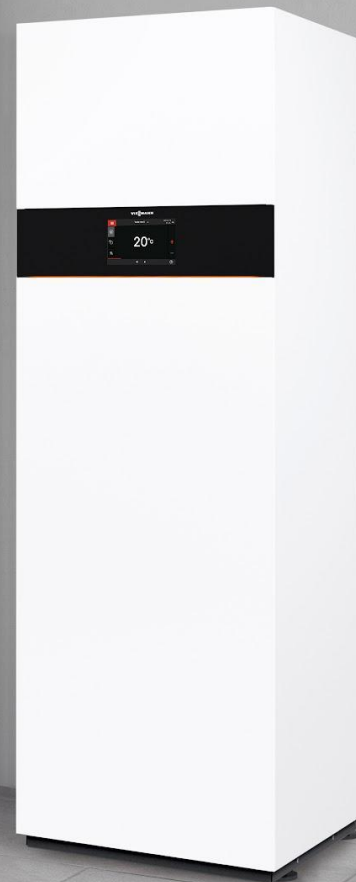


500,000 t CO₂

additional savings per year

Climate solutions

Serving an essential
human need in the
most efficient way



The Integrated Viessmann Solution Offering

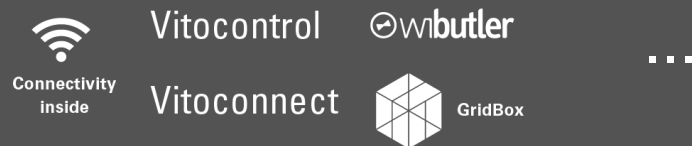
Value Added Services



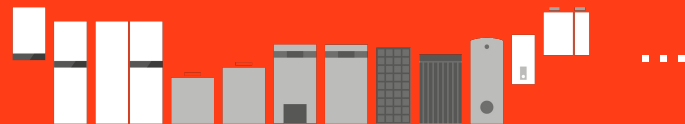
Digital Services



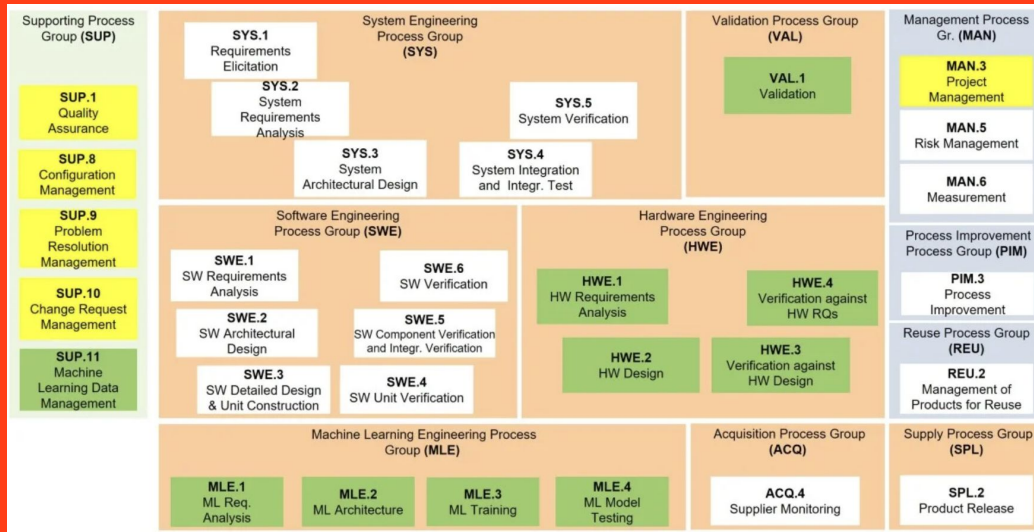
Connectivity & Platforms



Products & Systems



Verification and Validation



Quality Assurance

Quality Engineering

SUP.1 Quality Assurance

The purpose of the Quality Assurance process is to provide independent and objective assurance that work products and processes comply with defined criteria and that non-conformances are resolved and further prevented.

Why discuss KPIs, metrics and dashboards **with me?**

- Dashboards are my daily work
- I try to enable data visibility for project status for both sides
- My goal is to present overview on the level of quality for both sides



What KPIs, metrics and dashboards can give You?

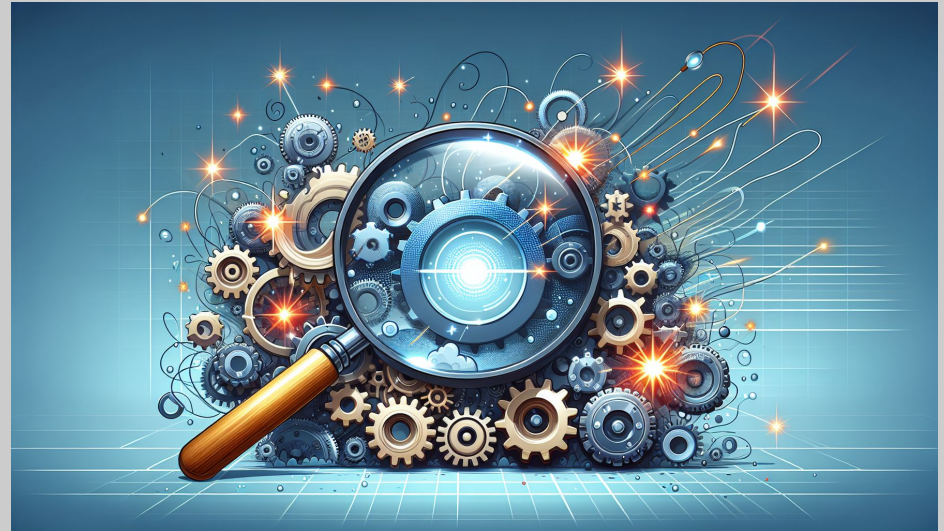
- Enable decisions from management
- Support the management with the insights and risks
- Support project initiatives enablement by teams
- Enable experiments and improvements



What is Quality?

What is quality definition by ISO?

“The capability of a software product to satisfy stated and implied needs when used under specified conditions.”



What is quality definition by IEEE?

“The degree to which those established requirements accurately represent stakeholder needs, wants, and expectations.”



Why is Quality important?

Technology
accelerates faster
and faster

Markets and business
models are under siege,

Who defends the Quality?



Before we go to metrics and KPIs, we must diagnose if they could help?

How company and its
culture affects quality?

Cultural elements that positively influence quality include:

- Excellence is valued over releasing quickly.
- There is pride in craftsmanship.
- Management support doing things the right way.
- Ethics and accountability are emphasized over profits or schedules.
- High quality work is rewarded and recognized. Quality is everyone's responsibility.
- Learning and improvement is promoted
- Collaboration is encouraged
- Customer satisfaction is seen as a key measure of quality.



Organizations can also have negative cultural influences:

- Lack of leadership commitment to quality.
- Underinvestment in training, tools, and processes.
- Conflicts between engineering and business goals.
- Poor communication and team dynamics.
- Frequent overtime indicates schedule pressure.
- Lack of transparency.
- Blame culture



How to measure and improve
quality of the deliveries?

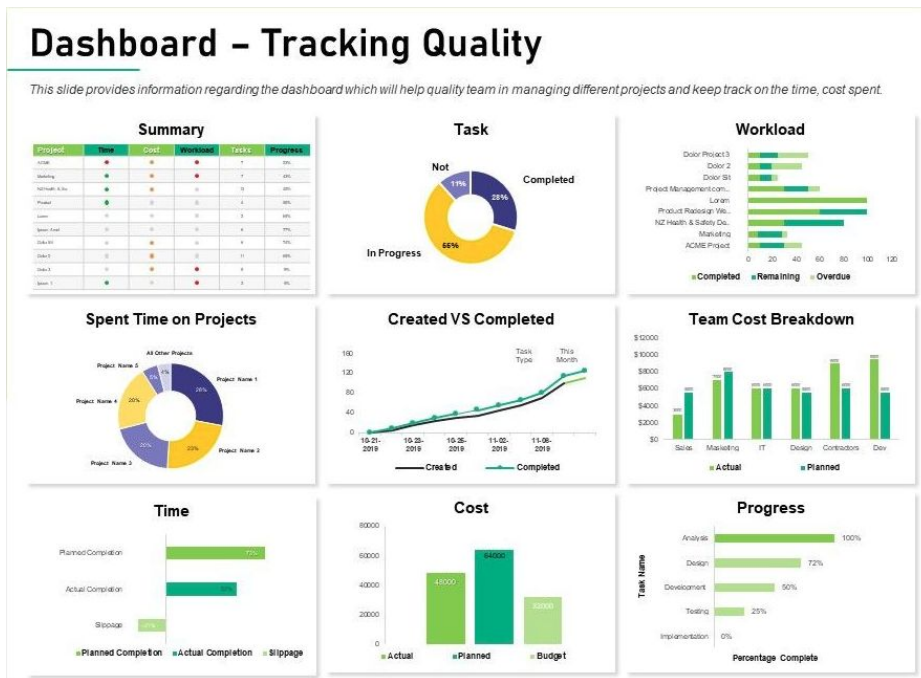
KPIs, metrics and dashboards

Quality initiatives for Software Development

Measurements are used to:

- monitor trends,
- compare benchmarks,
- predict failures,
- optimize testing,
- prioritize improvements.

Statistical analysis like Pareto charts helps interpret the data.



Metrics enable
Quality Management

Benefits of Quality Management (IEEE)?

- **Quality planning** – Defining quality objectives, requirements, targets, and planning of quality assurance activities.
- **Quality control** – Techniques to measure quality characteristics, review work products, and find defects.
- **Quality engineering** – Processes and audits to ensure compliance with procedures.
- **Quality improvement** – Defect analysis and process enhancements to improve quality.
- **Resources** – Infrastructure, tools, training that enable quality processes.
- **Standards** – Regulations, models, certifications that guide quality work.
- **Culture** – Values, behaviors that encourage quality mindset.

Best practices, For successful dashboards

Minimize the toolset and “automate everything”

- Minimize manual work !!
- Integrate with project management tool
 - JIRA->Confluence is good
 - JIRA integrated is better
- Adapt the tools to KPIs, not metrics to tools
- Invest in the interest of Leadership (show them the benefits) to get better understanding when requests on flow or parameters change got escalated
- Project focused dashboard with multiple metrics and controller to switch projects
- Top level dashboards with few metrics on cross projects or department view



**KEEP
CALM
AND
AUTOMATE
EVERYTHING**

Build dashboards and metrics to support 2 dimensions

Short term:

- Build a dashboard and set for ad-hoc current situation
- Support the release notes verification with metrics
- Enable ad hoc view in metrics by leadership team
- Plan and organize scheduled reviews i.e. monthly
- Raise flag and ask for quick improvements responding to trends



Build dashboards and metrics to support 2 dimensions

Long Term:

- Goals for project to achieve
- Tracking long term initiatives impact
- Measure the way toward the goals
- Enable monitoring by the team themselves
- Summarize the progress in i.e. yearly updates
- Be patient and watch trends
- Do not escalate long term goals ad-hoc



How to handle metrics with teams?

- Enable option to push additional metrics or allow them to be not applicable
- Do to enforce changes in the process for dashboard
- Give action to create dashboard and secure frequent syncs – observe progress in building and then review at the same meetings
- Do not force the priority, let the team to learn benefit from it and build priority themselves
- Review boards in few days advance and mark points to discuss (I try to keep 2 days advance)
- Review last sync action statuses
- Keep the review with the team short (I use 5-10 mins of 1hour booked)

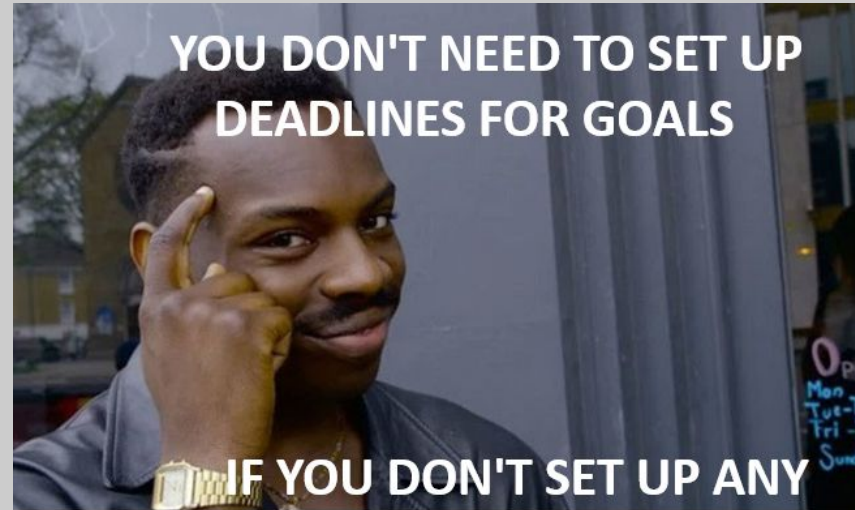
Who should use metrics?

- The Delivery Team
- The Project and Product Management
- Organizational Management
- Project and Product Stakeholders
- Everyone else who is interested



When is the time for metrics?

- There is no worse or better project stage to build or use dashboards
- Use it for presenting situation to project management
- Use it to build or review root causes to execute changes or decisions by management
- Use it as justification by the team when they try to escalate or trigger a change anywhere outside of the team
- Frequent reviews with team, with management and with stakeholders



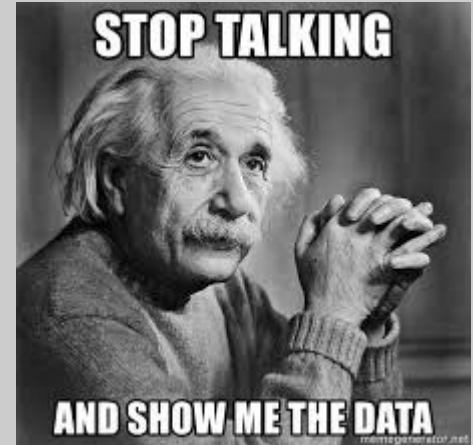
Goals of Quality Management (IEEE)

- Automating tedious tasks like testing, reviews, inspections. This boosts team efficiency
- Performing sophisticated analysis like structural code analysis that is difficult manually. Provides deeper insights
- Visualizing metrics, trends, and quality data. Improves focus for teams and managers
- Enforcing standards, best practices, architecture rules. Promotes consistency
- Supporting distributed teams and code reuse. Improves coordination
- Reducing risks associated with manual processes like human error, subjectivity, and oversight
- Capturing quality data to guide improvement initiatives. Provides process feedback
- Integrating quality practices within IDEs to provide developer support

Minimal core set?
(PoC scope)

What sounds great for management, but not always is easy to be monitored..

- **Error density** – errors per size of work product (requirements, design, code).
- **Defect density** – defects per size of software.
- **Failure rate** – mean time between failures.
- **Reliability models** – estimate future failures from defect data.
- **Escaped defects** – defects missed during development found by others



Our favourites - defect related

(outside of simple priority and severity split)

1. Defect discovery

- a. Bugs inside component
- b. Bugs found in other components
- c. Cross component issues

2. **Bug release** - stats of defects with risk assessment accepted to be introduced inside the release of component or system to the field

3. **Bug escape** - numbers of defects that return from the field with additional stats split.

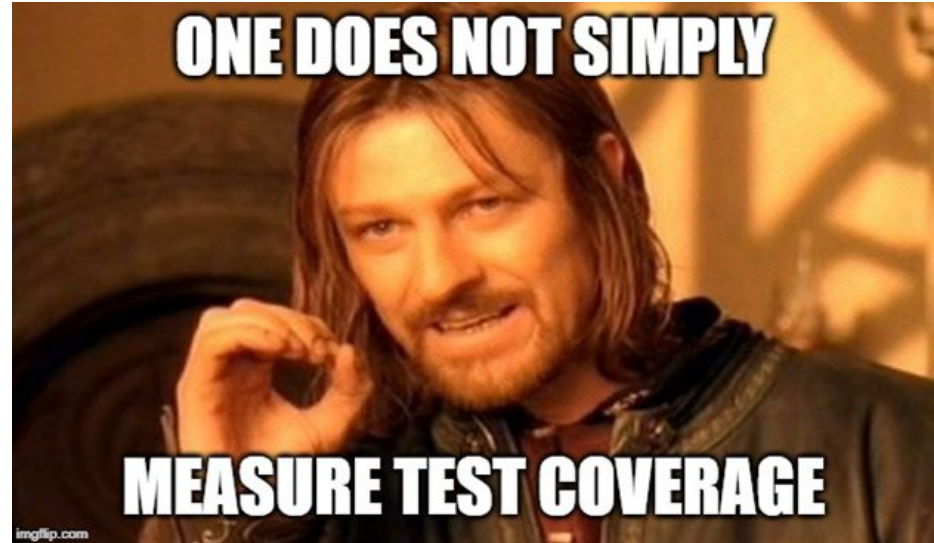


Our favourite - coverage related

We love to analyze all the base traceability practices from ASPICE

1. **Code Coverage** and internal/private interfaces - line or statements for units testing
2. **Architecture/Pubic interfaces coverage** on integration testing
3. **Requirements coverage** on functional testing

Of course all in the scope supported by technology and tools used in the project.



Of course there might be
more..

Your creativity is the limit, or maybe the tools, but few examples:

- Project level (Reqs, Architecture, technical docs, roadmap etc.)
- Scrum level (DoR, DoD, etc.)
- Operation (CI/CD, JIRA, Test Management)
- Open issues backlog
- Bugs trends short and long term
- Internal vs external errors
- High level automation coverage
- Low level dev verification coverage
- Manual test results
- Release notes
- Requirements status
- Backlog delivery status
- Project initiatives
- UI/UX definitions
- CRs, production issues injection
- Accepted vs Rejected bugs comparison
- Static code analysis
- Etc.

Summarize:

- Way to present data
- Make them integrated
- **Automate everything**
- **Make justified decisions**
- Measure impact of initiatives
- **Measure the Quality of Delivery**
- Discuss and present current situation
- **Plan for the future**



Research Your tools and
build You own dashboards !!



Any Questions?

Thank You and have fun !!



VIESMANN