TNG TECHNOLOGY CONSULTING

# Scrum in Large Projects

## Theory and Practice

Big Techday 5
Munich, June 15, 2012
Dr. Sebastian Stamminger

TNG TECHNOLOGY CONSULTING

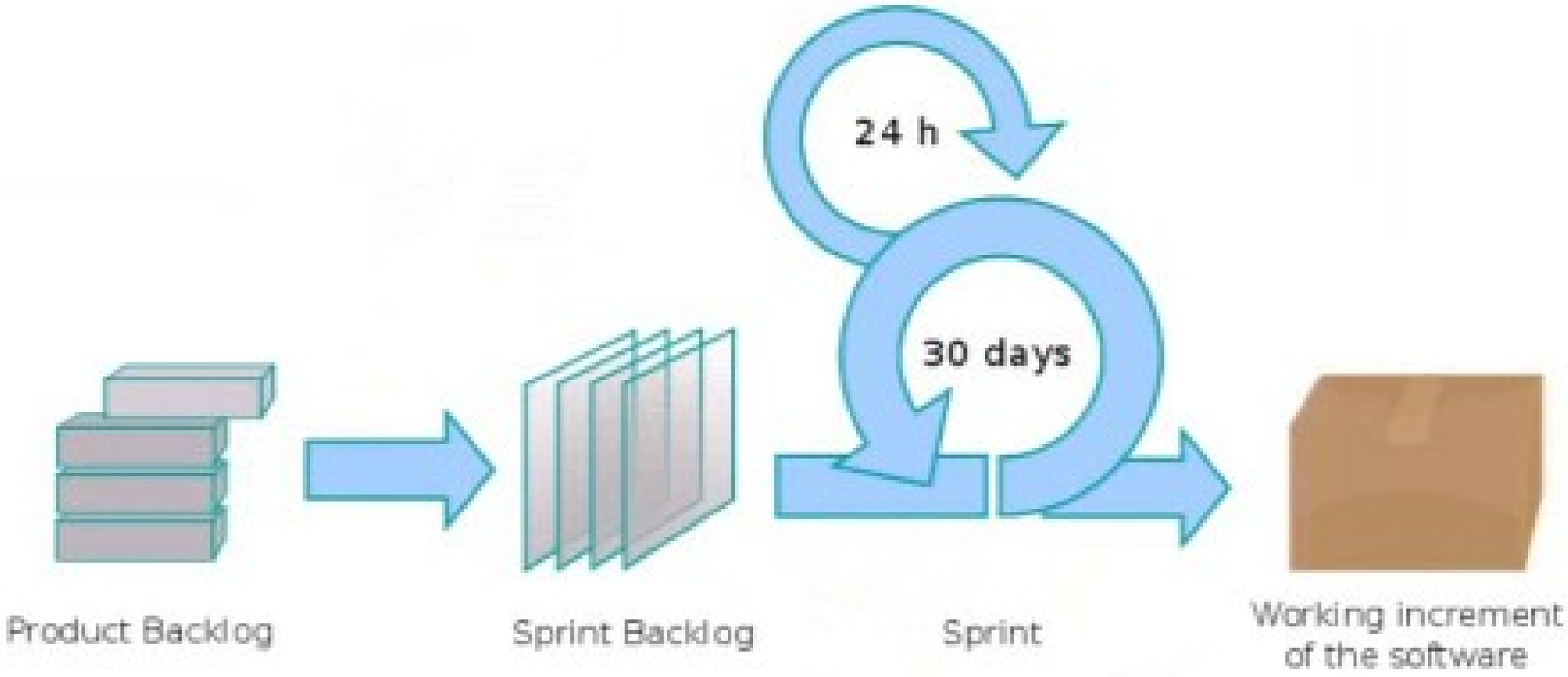
# Agenda

- Theory
- Case Study
- Teams
- Our Process
- Challenges
- Lessons Learned

TNG TECHNOLOGY CONSULTING

## Scrum

- Basic knowledge of Scrum is assumed to be known
- In this talk: extensions of Scrum for large projects



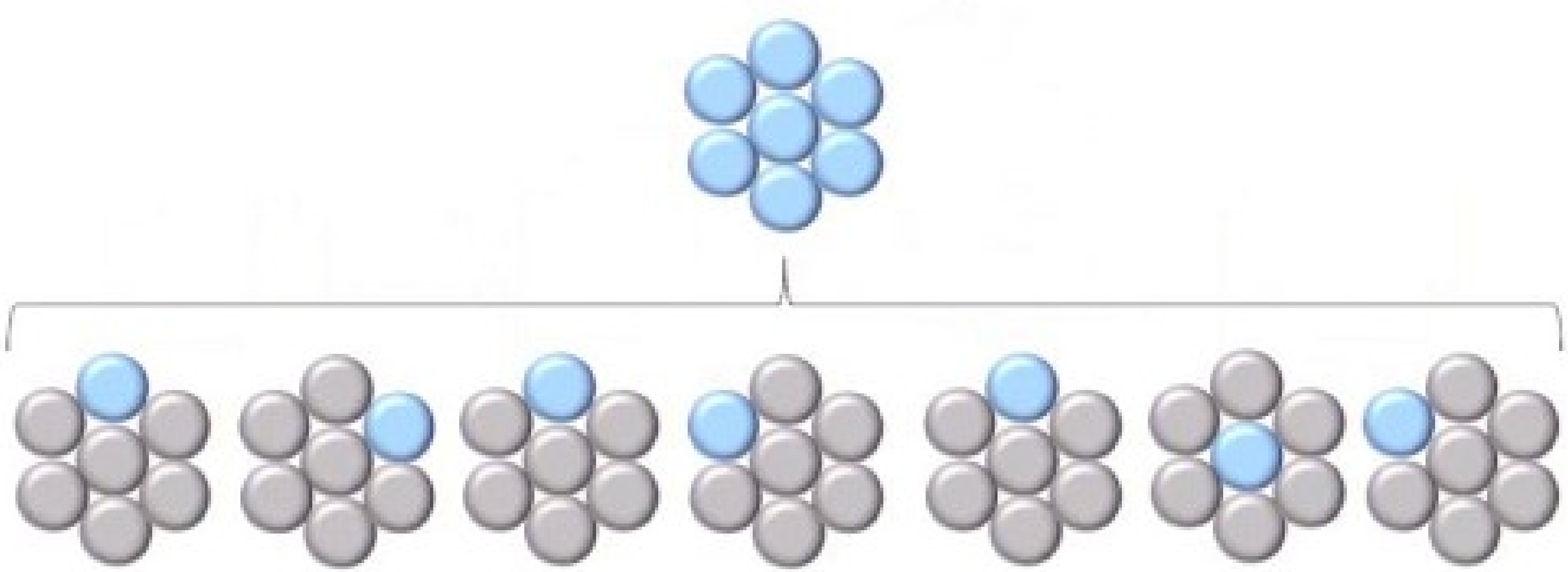

Source: Scrum process.svg, Creative Commons by-sa-3.0, Author: Lakeworks

TNG TECHNOLOGY CONSULTING

## Scrum of Scrums

- One representative of every team
- Daily standup meeting
- Goal: coordination, self-organization

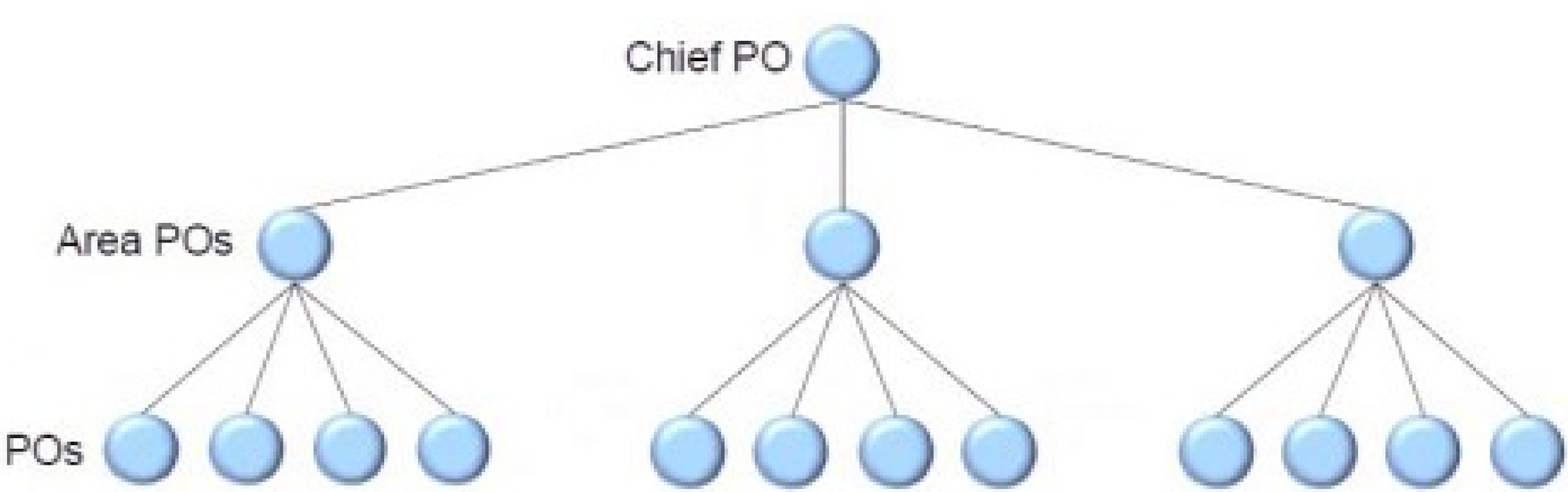
- Also: PO Daily, SM Daily
- Coordination of the POs, and the SMs



TNG TECHNOLOGY CONSULTING

## Requirement Areas

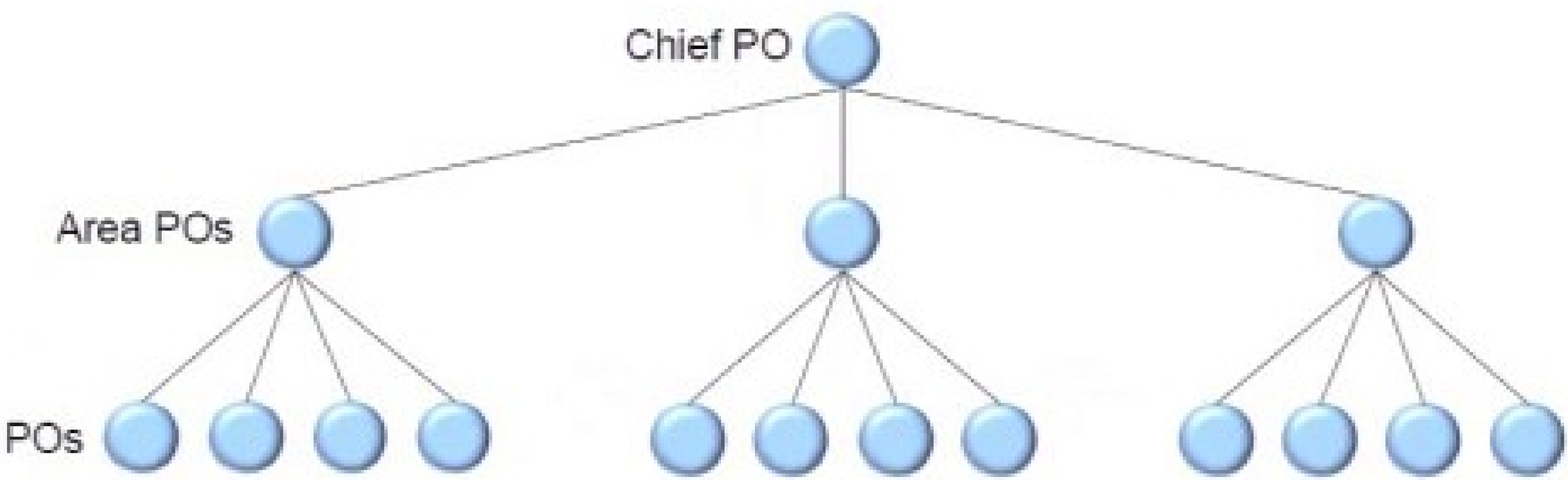
- If there are too many requirements for one backlog, i.e. none can survey and prioritize them, they should to be grouped into so called **requirement areas**
- Area PO responsible for the area backlog
- Hierarchy of POs



TNG TECHNOLOGY CONSULTING

## Requirement Areas

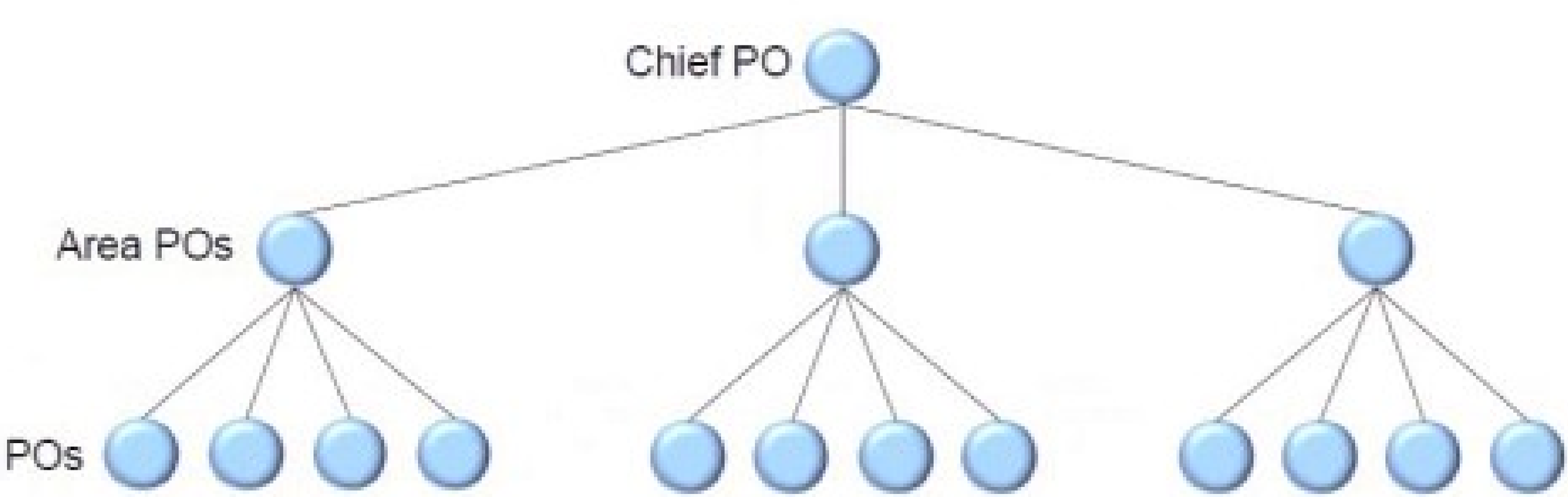
- If there are too many requirements for one backlog, i.e. none can survey and prioritize them, they should to be grouped into so called **requirement areas**
- Area PO responsible for the area backlog
- Hierarchy of POs



TNG TECHNOLOGY CONSULTING

## Requirement Areas

- If there are too many requirements for one backlog, i.e. none can survey and prioritize them, they should to be grouped into so called **requirement areas**
- Area PO responsible for the area backlog
- Hierarchy of POs



TNG TECHNOLOGY CONSULTING

## Component Teams vs. Feature Teams

- Component team: responsible for one system component

- Feature team
  - cross-functional
  - responsible for complete features
  - across several components

TNG TECHNOLOGY CONSULTING

## Component Teams vs. Feature Teams

- Component team: responsible for one system component

- Feature team
  - cross-functional
  - responsible for complete features
  - across several components

## Component Teams vs. Feature Teams

- Component team: responsible for one system component

- Feature team
  - cross-functional
  - responsible for complete features
  - across several components

- Pros and cons of feature teams (compared to component teams)
  - Pros
    - focus on business value
    - less dependencies
    - less waste (waiting, coordination, unused components)
    - increased learning, better code/design quality, higher motivation
  - Cons
    - potentially unclear responsibility for components or subsystems
    - increased learning (new components, new people)
    - may require organizational change

## Component Teams vs. Feature Teams

- Component team: responsible for one system component

- Feature team
  - cross-functional
  - responsible for complete features
  - across several components

- Pros and cons of feature teams (compared to component teams)
  - Pros
    - focus on business value
    - less dependencies
    - less waste (waiting, coordination, unused components)
    - increased learning, better code/design quality, higher motivation
  - Cons
    - potentially unclear responsibility for components or subsystems
    - increased learning (new components, new people)
    - may require organizational change

## Component Teams vs. Feature Teams

- Component team: responsible for one system component

- Feature team
  - cross-functional
  - responsible for complete features
  - across several components

- Pros and cons of feature teams (compared to component teams)
  - Pros
    - focus on business value
    - less dependencies
    - less waste (waiting, coordination, unused components)
    - increased learning, better code/design quality, higher motivation
  - Cons
    - potentially unclear responsibility for components or subsystems
    - increased learning (new components, new people)
    - may require organizational change

## Support Teams

- Cross-team tasks can be delivered by support teams, e.g.
  - infrastructure, build environment, staging environment
  - architecture evaluations
  - business concepts/strategy

- Goal:
  - support the feature teams
  - not: give them directives

## Communities of Practice

- Sometimes also called Virtual Teams
- Know-how exchange across teams on e.g.
  - architecture
  - certain technologies
  - concepts
  - methodologies
- ideally self-organized (e.g. as result of a retrospective)
- voluntary participation

- The organization can encourage building communities of practice

## Communities of Practice

- Sometimes also called Virtual Teams
- Know-how exchange across teams on e.g.
  - architecture
  - certain technologies
  - concepts
  - methodologies
- ideally self-organized (e.g. as result of a retrospective)
- voluntary participation

- The organization can encourage building communities of practice

## Joint Review, Retrospective

- Joint Review
  - for all project members or just the POs
  - presentation of the features developed in the last sprint
  - goal: spread knowledge, fascination for the whole product

- Joint Retrospective
  - for all project members or
  - for representatives, e.g.
    - Virtual Teams
    - Scrum Masters
    - Product Owners
  - goal: improve the whole project, not only single teams

## Joint Review, Retrospective

- Joint Review
  - for all project members or just the POs
  - presentation of the features developed in the last sprint
  - goal: spread knowledge, fascination for the whole product

- Joint Retrospective
  - for all project members or
  - for representatives, e.g.
    - Virtual Teams
    - Scrum Masters
    - Product Owners
  - goal: improve the whole project, not only single teams

## Joint Review, Retrospective

- Joint Review
  - for all project members or just the POs
  - presentation of the features developed in the last sprint
  - goal: spread knowledge, fascination for the whole product

- Joint Retrospective
  - for all project members or
  - for representatives, e.g.
    - Virtual Teams
    - Scrum Masters
    - Product Owners
  - goal: improve the whole project, not only single teams

## Joint Review, Retrospective

- Joint Review
  - for all project members or just the POs
  - presentation of the features developed in the last sprint
  - goal: spread knowledge, fascination for the whole product

- Joint Retrospective
  - for all project members or
  - for representatives, e.g.
    - Virtual Teams
    - Scrum Masters
    - Product Owners
  - goal: improve the whole project, not only single teams

TNG TECHNOLOGY CONSULTING

# Agenda

- Theory
- **Case Study**
- Teams
- Our Process
- Challenges
- Lessons Learned

TNG TECHNOLOGY CONSULTING

## Project Setting

- Automotive company
- New development and replacement of an existing system
- Parts of the system are visible to the end customer
- Integration into existing system landscape, e.g.
  - CRM systems
  - vehicle data, financial data systems
  - dealer systems
- Huge amount of requirements

TNG TECHNOLOGY CONSULTING

## Project Setting

- Automotive company
- New development and replacement of an existing system
- Parts of the system are visible to the end customer
- Integration into existing system landscape, e.g.
  - CRM systems
  - vehicle data, financial data systems
  - dealer systems
- Huge amount of requirements

TNG TECHNOLOGY CONSULTING

## Project Setting

- Automotive company
- New development and replacement of an existing system
- Parts of the system are visible to the end customer
- Integration into existing system landscape, e.g.
  - CRM systems
  - vehicle data, financial data systems
  - dealer systems
- Huge amount of requirements

TNG TECHNOLOGY CONSULTING

## Project Setting

- Automotive company
- New development and replacement of an existing system
- Parts of the system are visible to the end customer
- Integration into existing system landscape, e.g.
  - CRM systems
  - vehicle data, financial data systems
  - dealer systems
- Huge amount of requirements

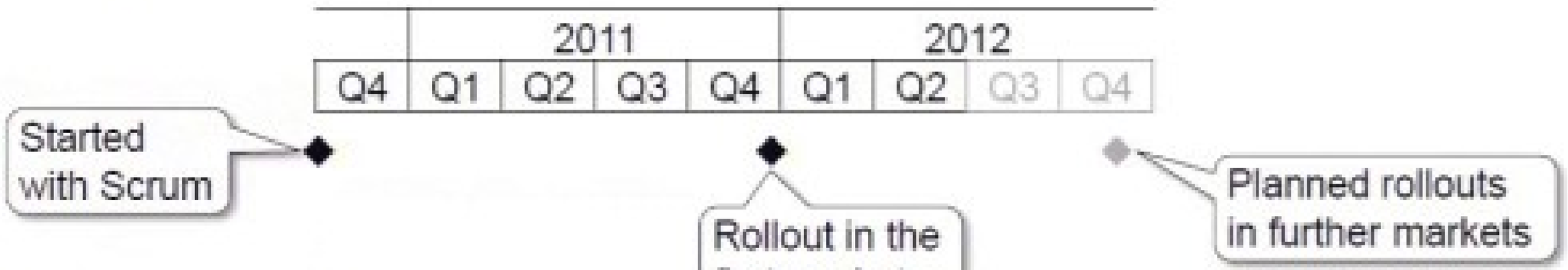
TNG TECHNOLOGY CONSULTING

## Timeline

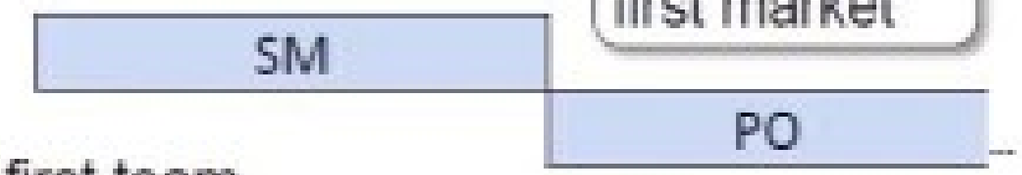
- Project got stuck in the requirements review phase after one year

  → experiment: Scrum (supported by top management)

- Timeline




- My role:




  - coaching the first team
  - SM for this team
  - then PO for a different team

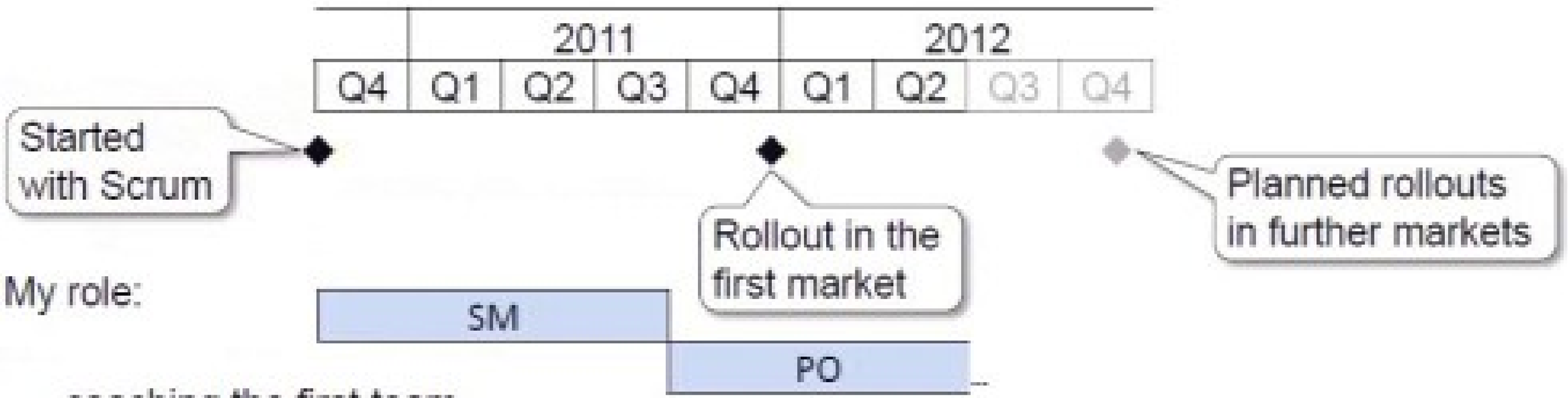
TNG TECHNOLOGY CONSULTING

## Timeline

- Project got stuck in the requirements review phase after one year

  → experiment: Scrum (supported by top management)

- Timeline




- My role:




  - coaching the first team
  - SM for this team
  - then PO for a different team

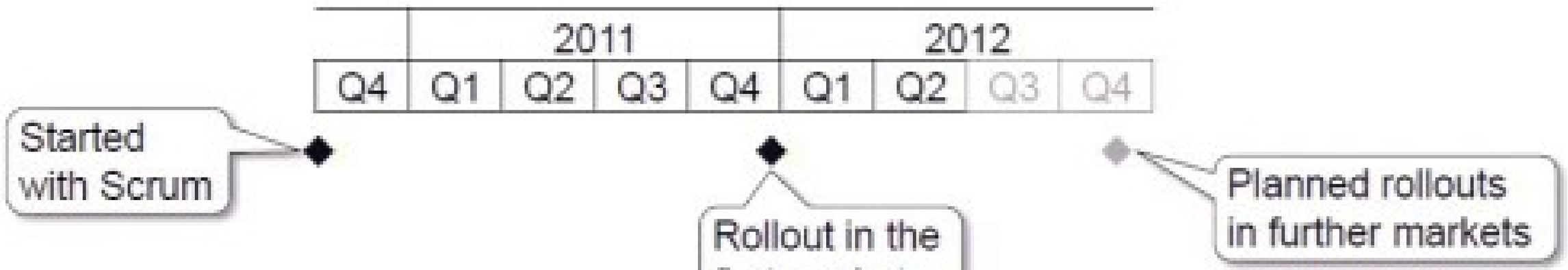
TNG TECHNOLOGY CONSULTING

## Timeline

- Project got stuck in the requirements review phase after one year

  → experiment: Scrum (supported by top management)

- Timeline




- My role:

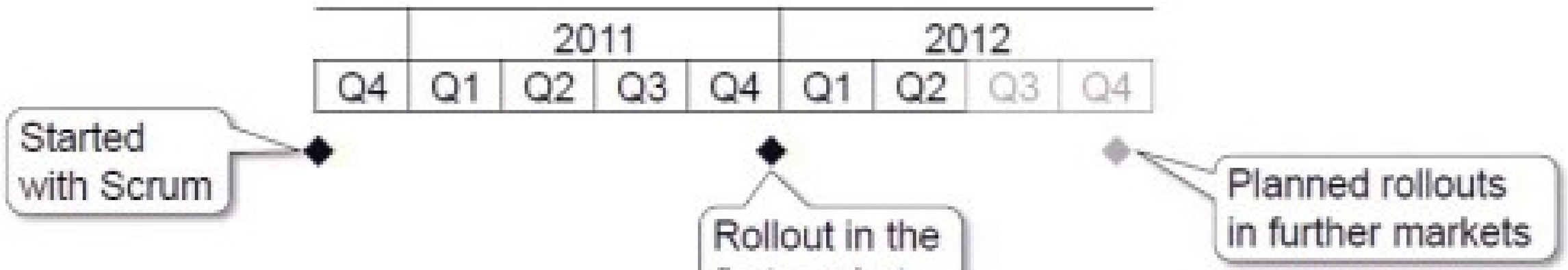

  - coaching the first team
  - SM for this team
  - then PO for a different team

## Timeline

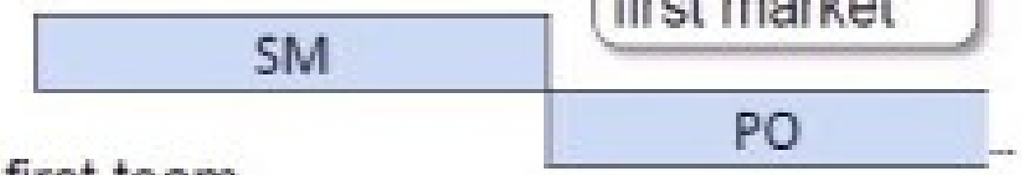
- Project got stuck in the requirements review phase after one year

  → experiment: Scrum (supported by top management)

- Timeline




- My role:




  - coaching the first team
  - SM for this team
  - then PO for a different team

TNG TECHNOLOGY CONSULTING

## Epics and Themes

- Wording:
  - theme = large user story
  - epic = huge user story

- The requirements were structured into
  - 11 epics, which were broken down into
  - 156 themes
  - assumption: 2 – 100 user stories per theme (in fact there were 7 on average)

- Identified in a few workshops with the customers

- Written on story cards and laid out on a table, see next slide

TNG TECHNOLOGY CONSULTING

## Epics and Themes

- Wording:
  - theme = large user story
  - epic = huge user story

- The requirements were structured into
  - 11 epics, which were broken down into
  - 156 themes
  - assumption: 2 – 100 user stories per theme (in fact there were 7 on average)

- Identified in a few workshops with the customers

- Written on story cards and laid out on a table, see next slide

TNG TECHNOLOGY CONSULTING

## Epics and Themes

- Wording:
  - theme = large user story
  - epic = huge user story

- The requirements were structured into
  - 11 epics, which were broken down into
  - 156 themes
  - assumption: 2 – 100 user stories per theme (in fact there were 7 on average)

- Identified in a few workshops with the customers

- Written on story cards and laid out on a table, see next slide

TNG TECHNOLOGY CONSULTING

## Epics and Themes

- Wording:
  - theme = large user story
  - epic = huge user story

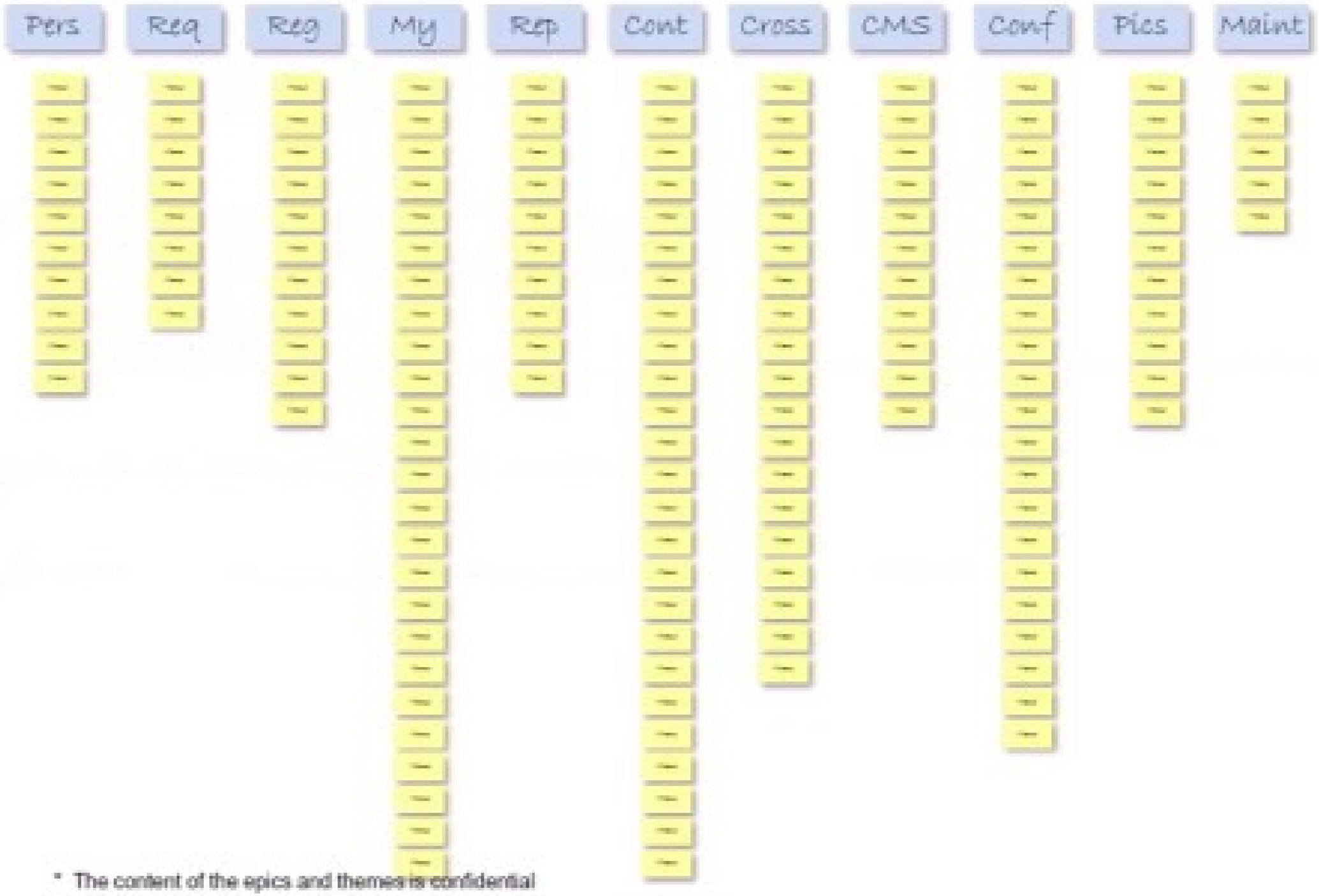
- The requirements were structured into
  - 11 epics, which were broken down into
  - 156 themes
  - assumption: 2 – 100 user stories per theme (in fact there were 7 on average)

- Identified in a few workshops with the customers

- Written on story cards and laid out on a table, see next slide

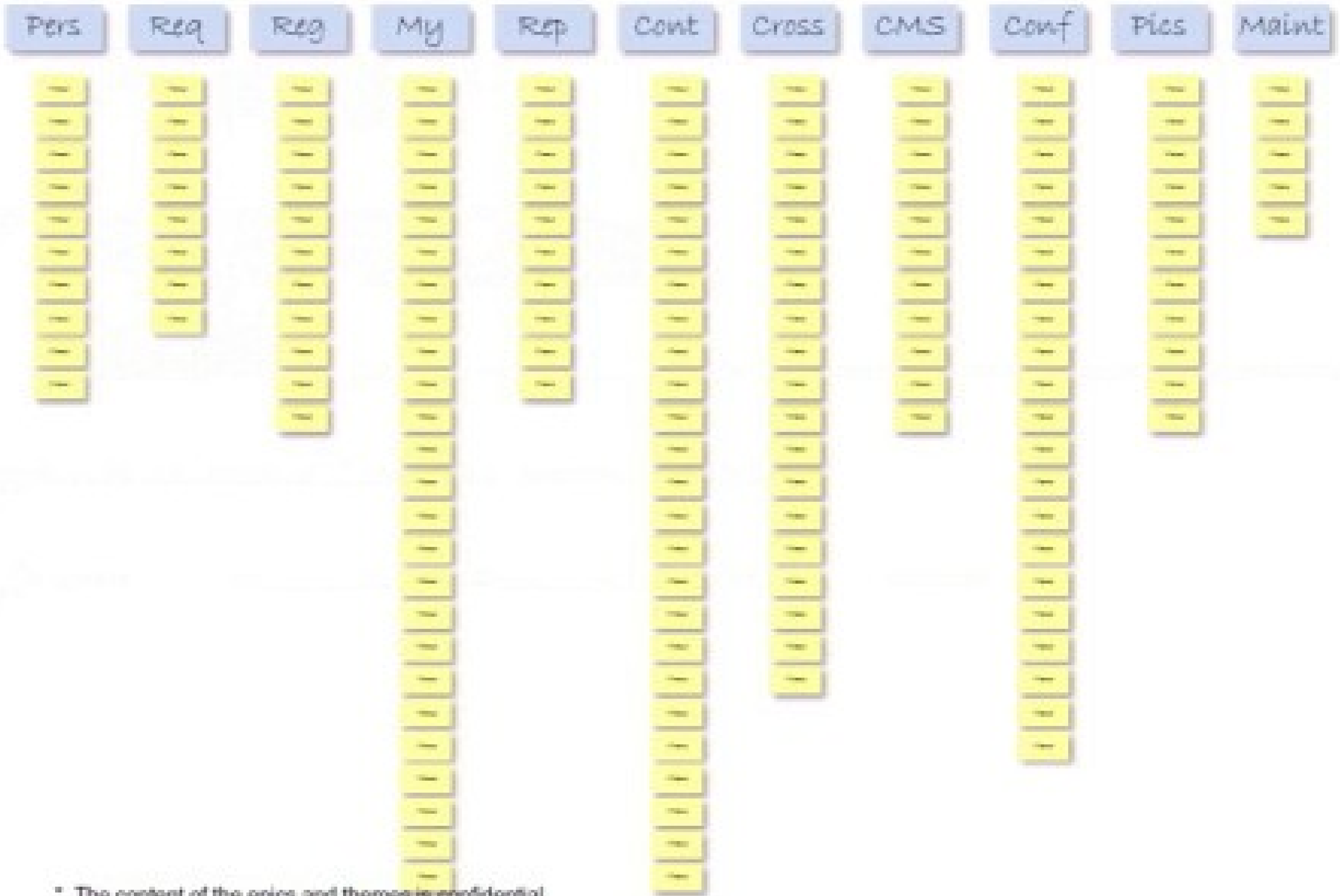
TNG TECHNOLOGY CONSULTING

# Epics and Themes



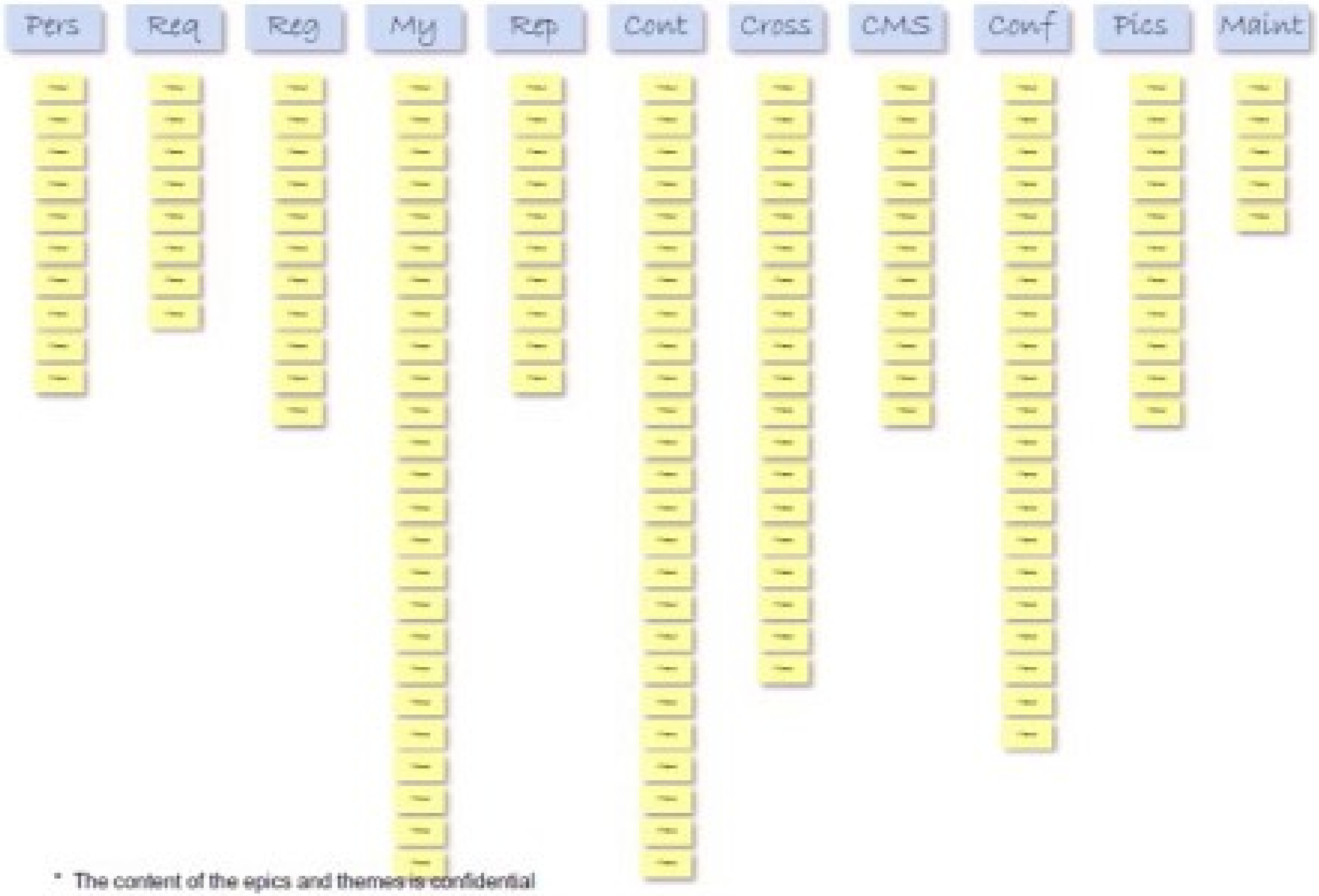

* The content of the epics and themes is confidential

# Epics and Themes




* The content of the epics and themes is confidential

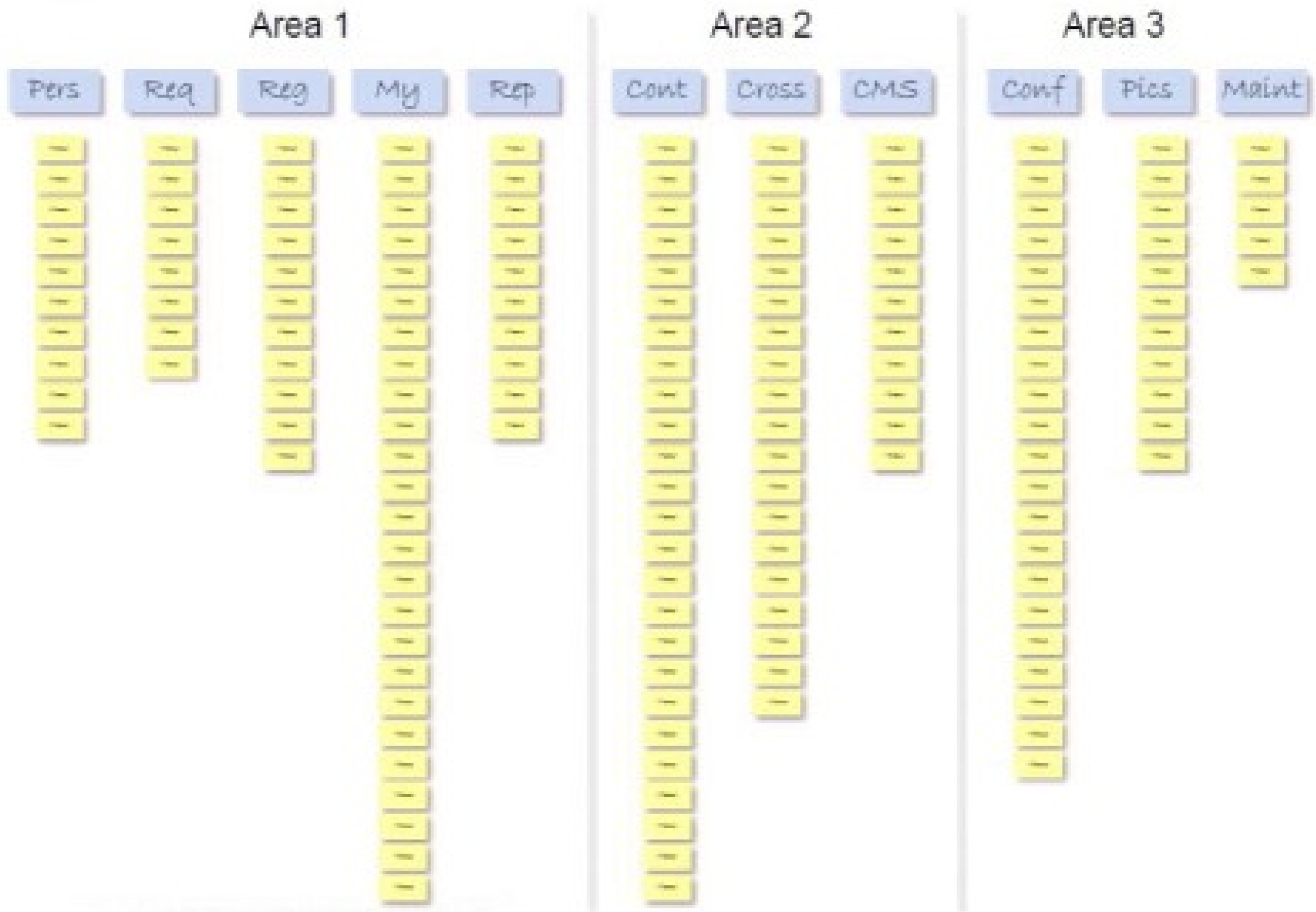
# Epics and Themes




* The content of the epics and themes is confidential

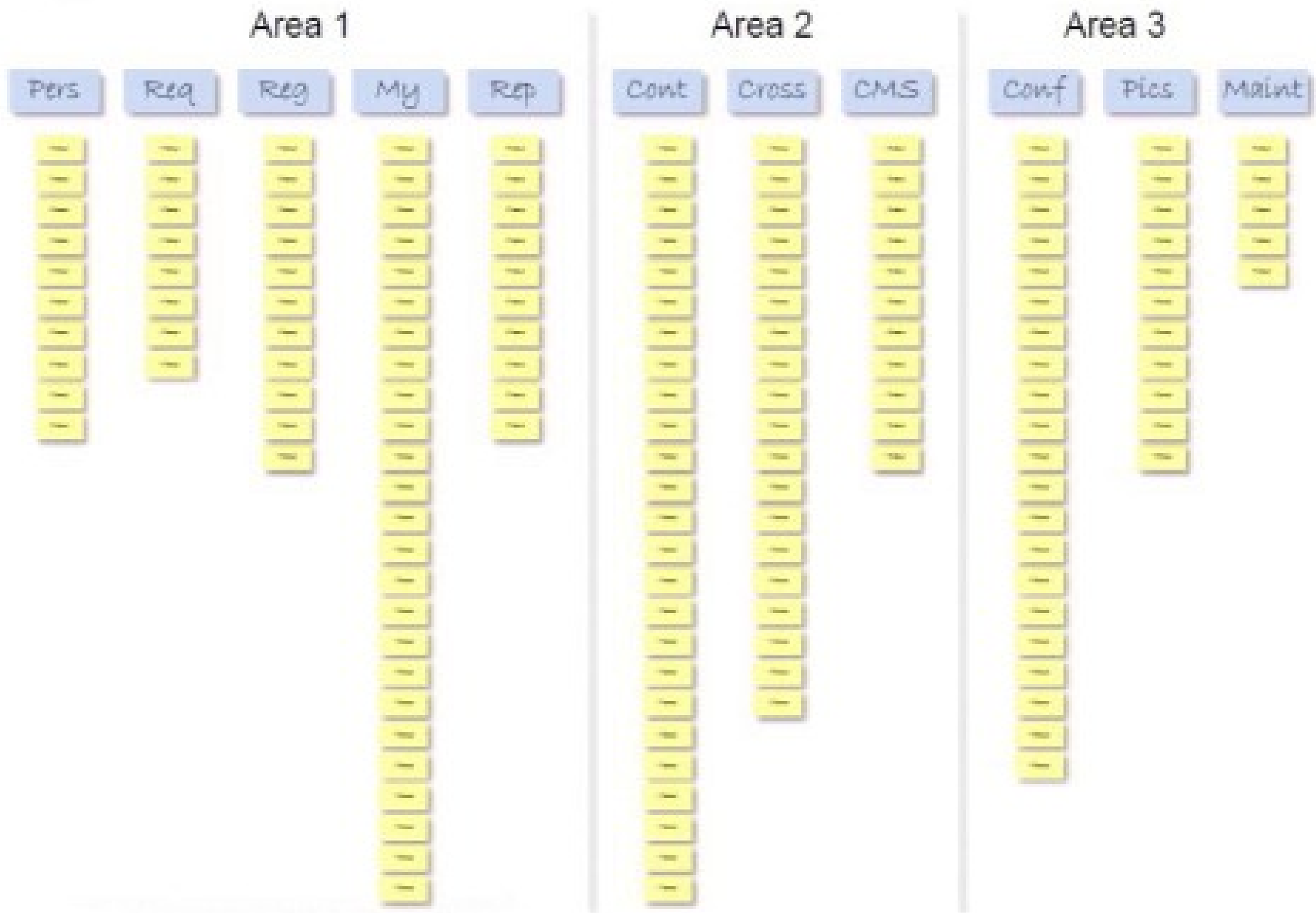
TNG TECHNOLOGY CONSULTING

## Requirement Areas



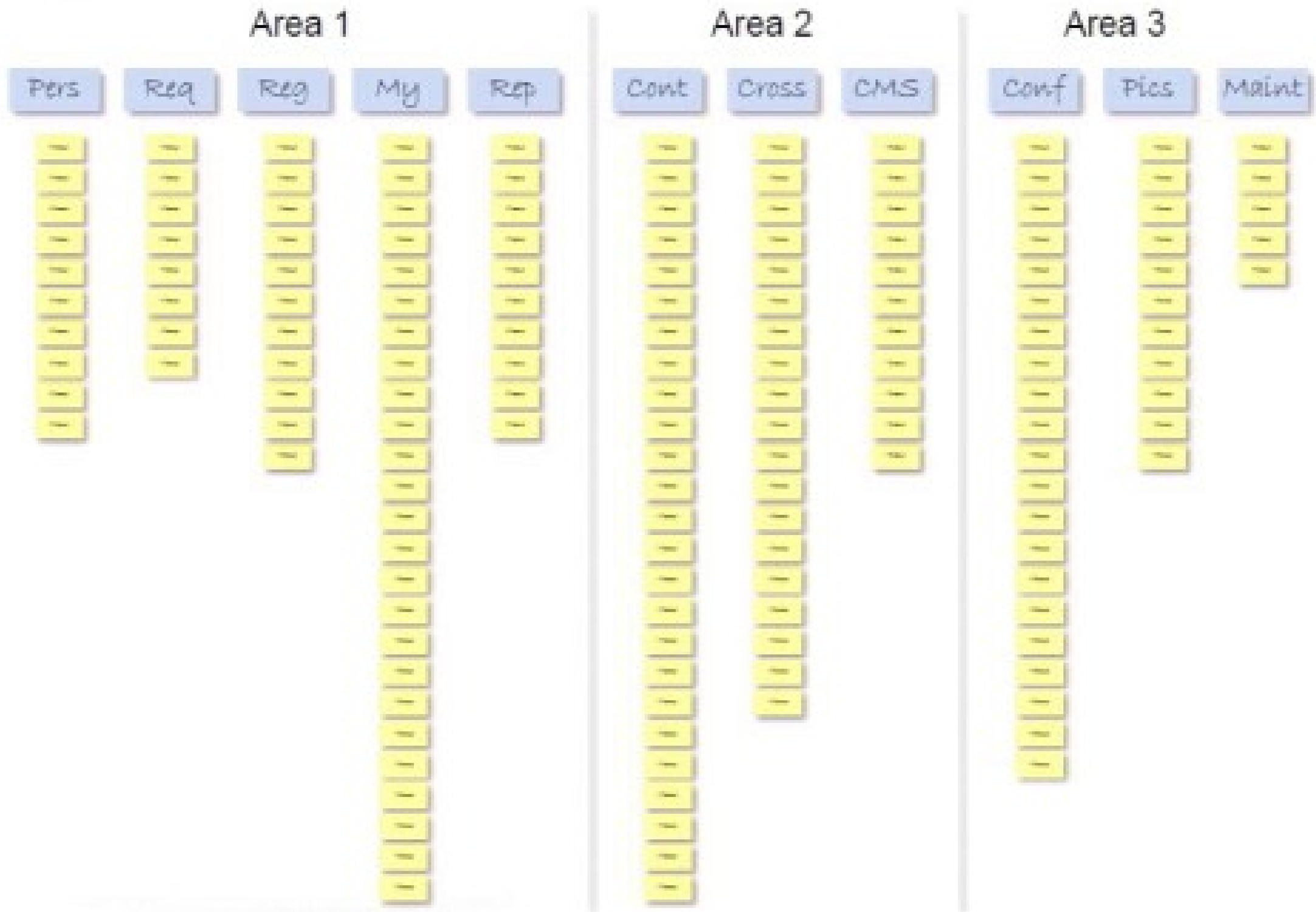
TNG TECHNOLOGY CONSULTING

## Requirement Areas



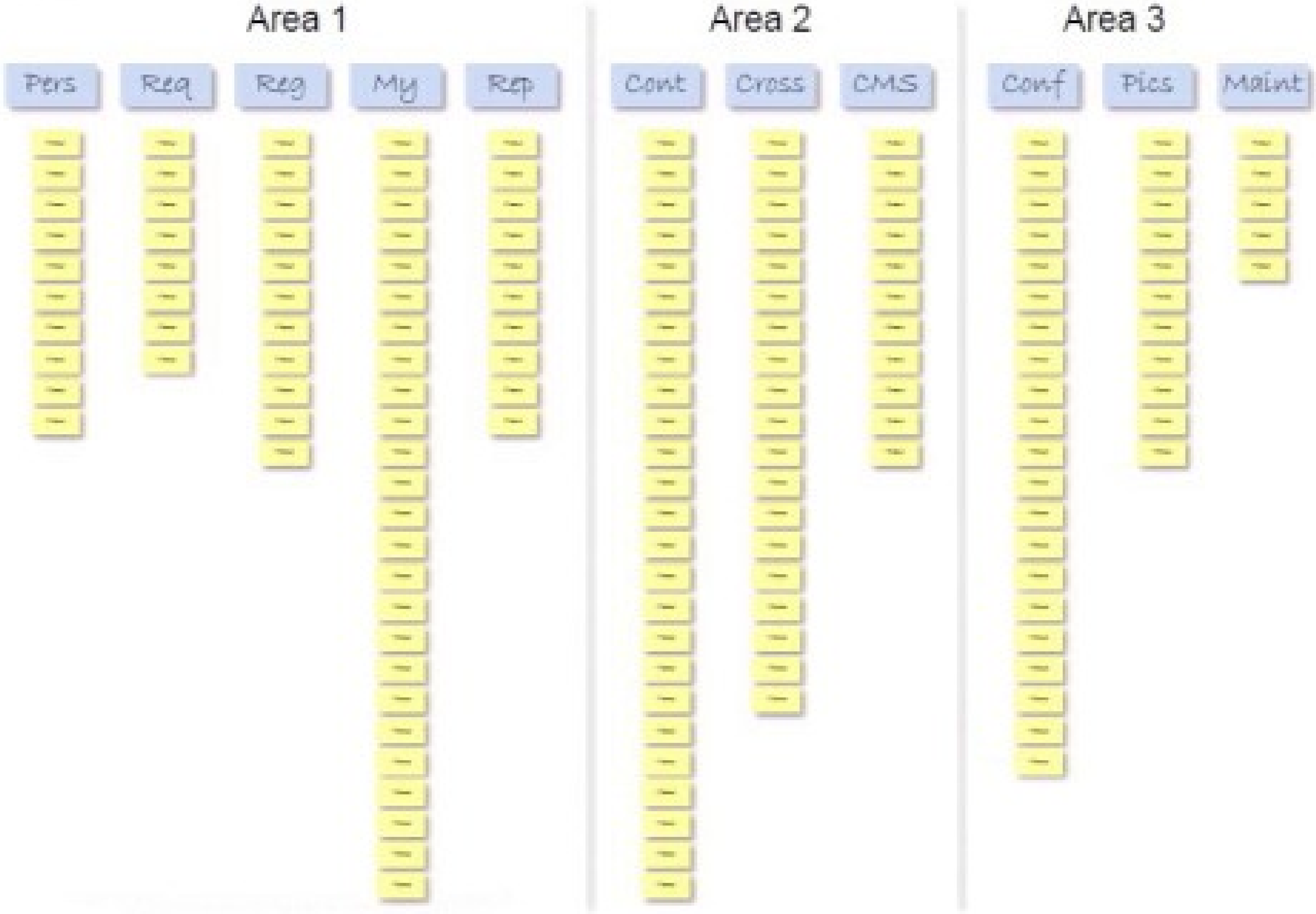
TNG TECHNOLOGY CONSULTING

## Requirement Areas



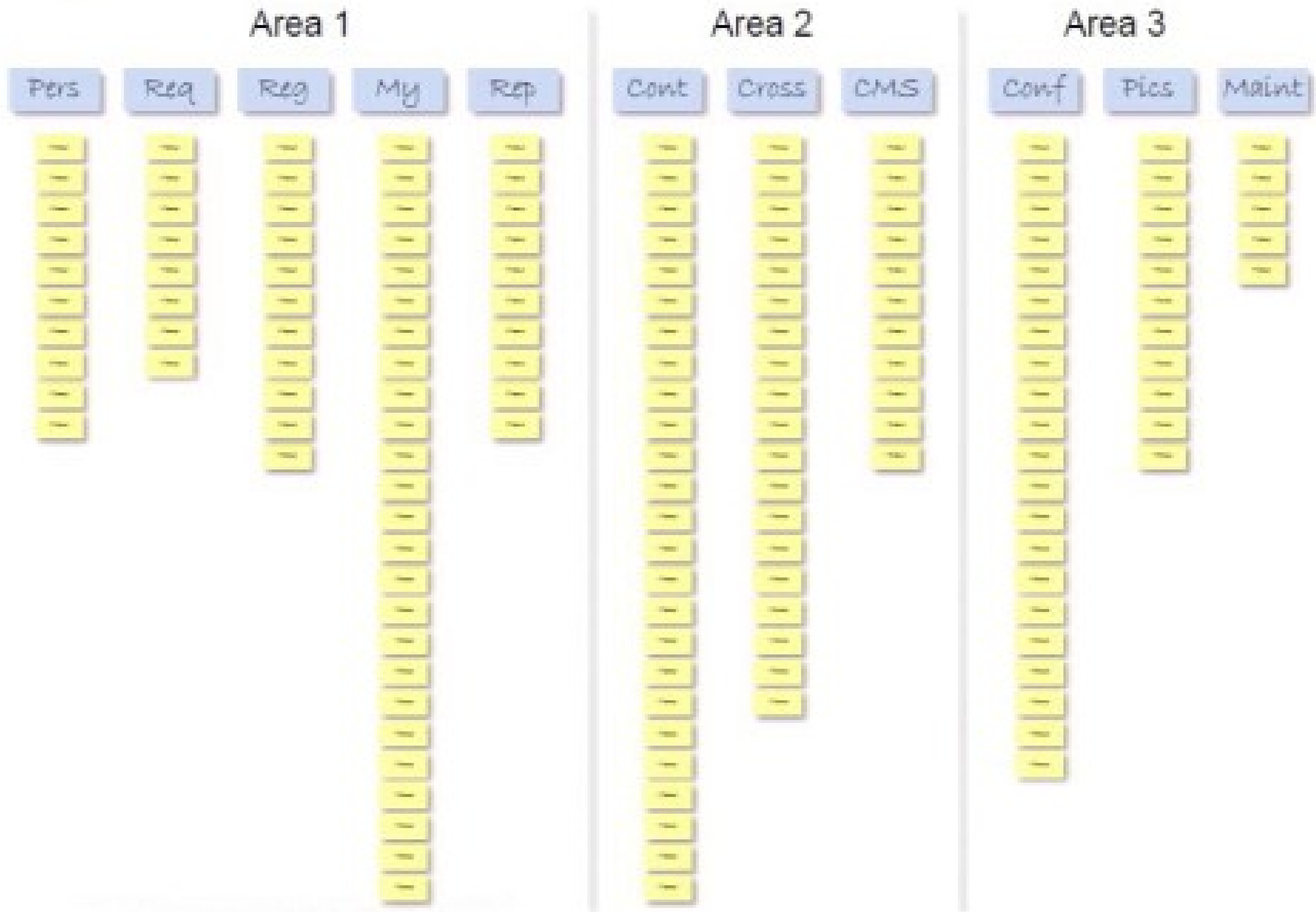
TNG TECHNOLOGY CONSULTING

## Requirement Areas



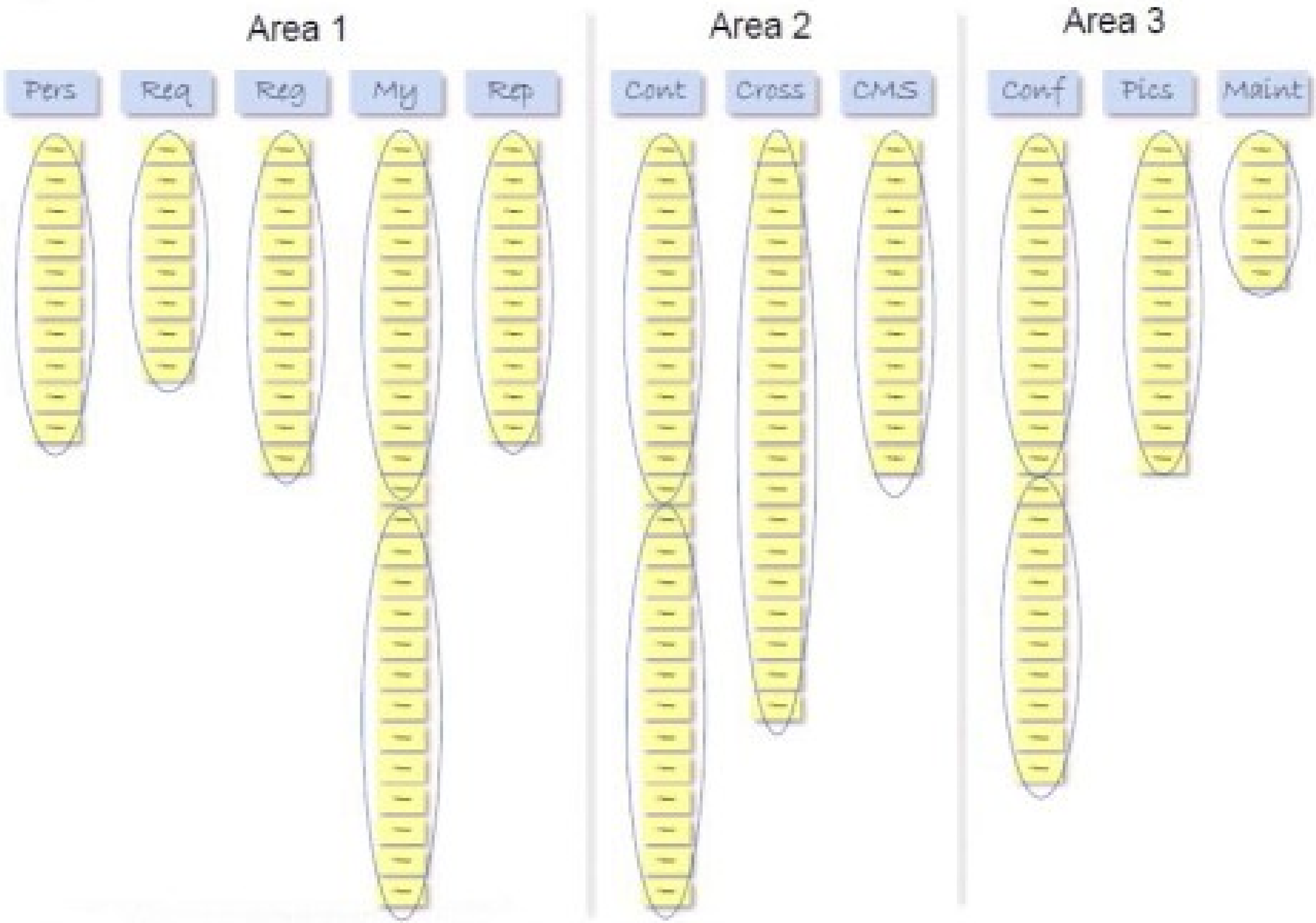
TNG TECHNOLOGY CONSULTING

## Requirement Areas



TNG TECHNOLOGY CONSULTING

## Epics and Teams

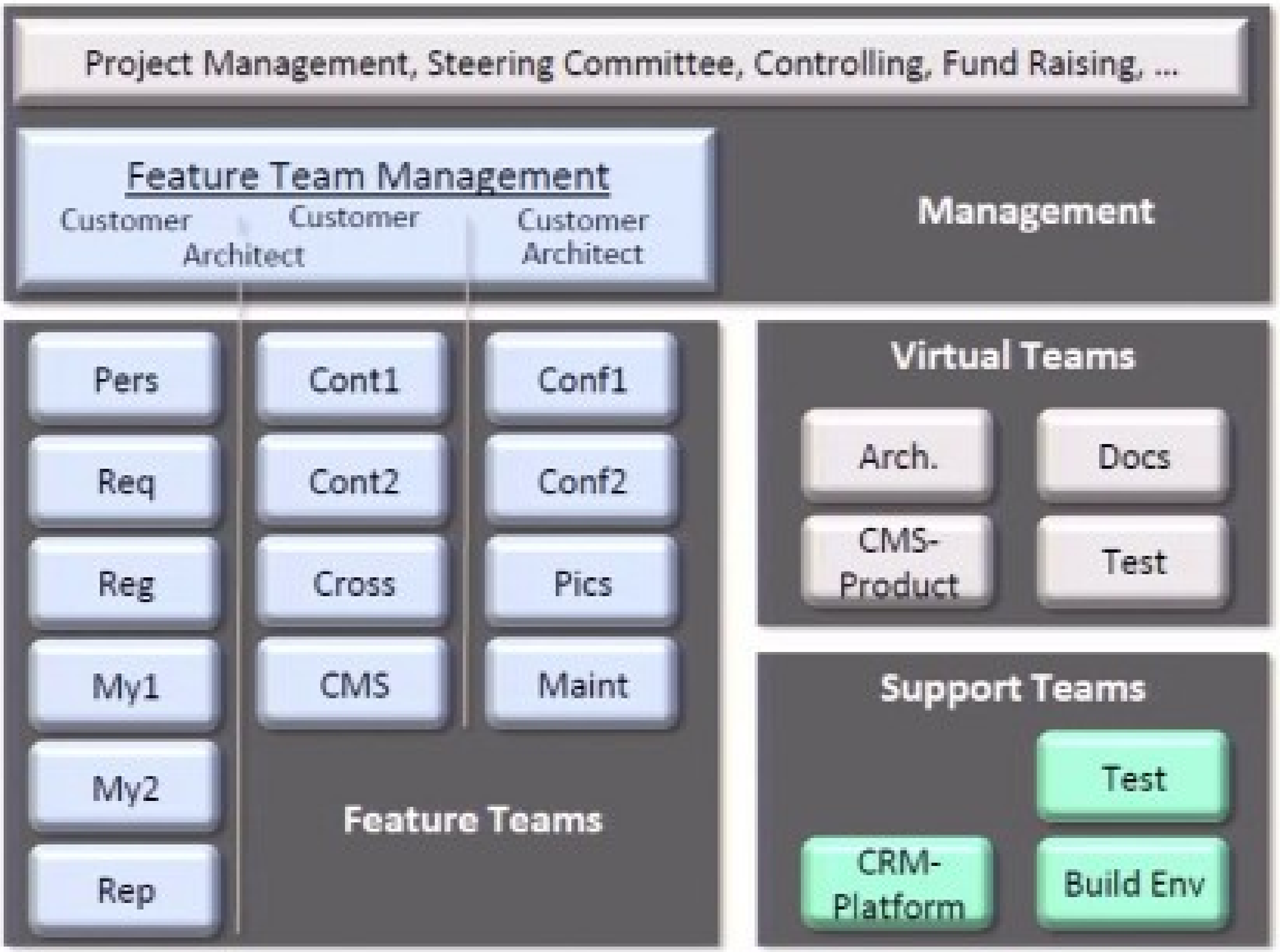


# Agenda

- Theory
- Case Study
- **Teams**
- Our Process
- Challenges
- Lessons Learned

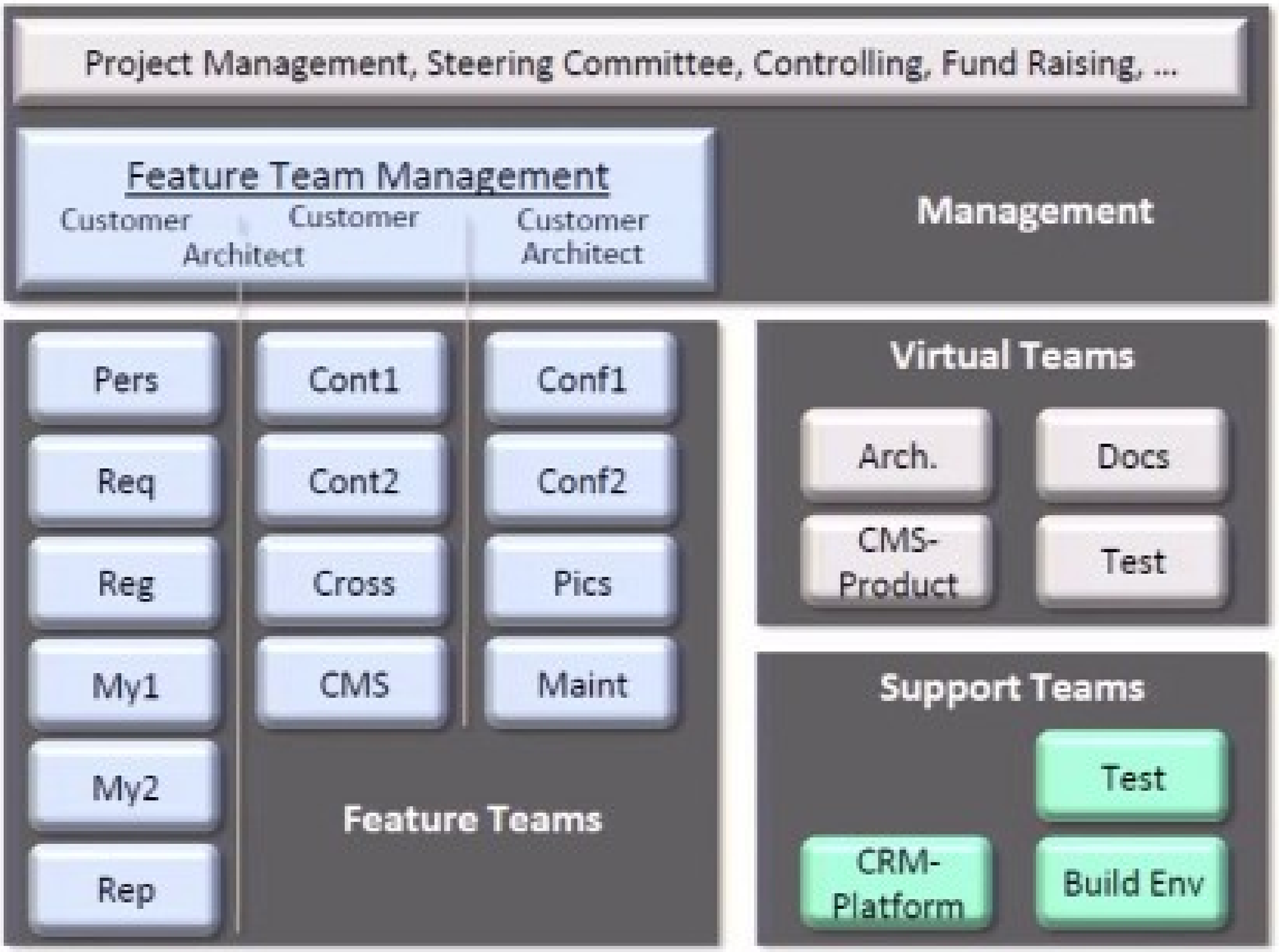
TNG TECHNOLOGY CONSULTING

# Organization Chart



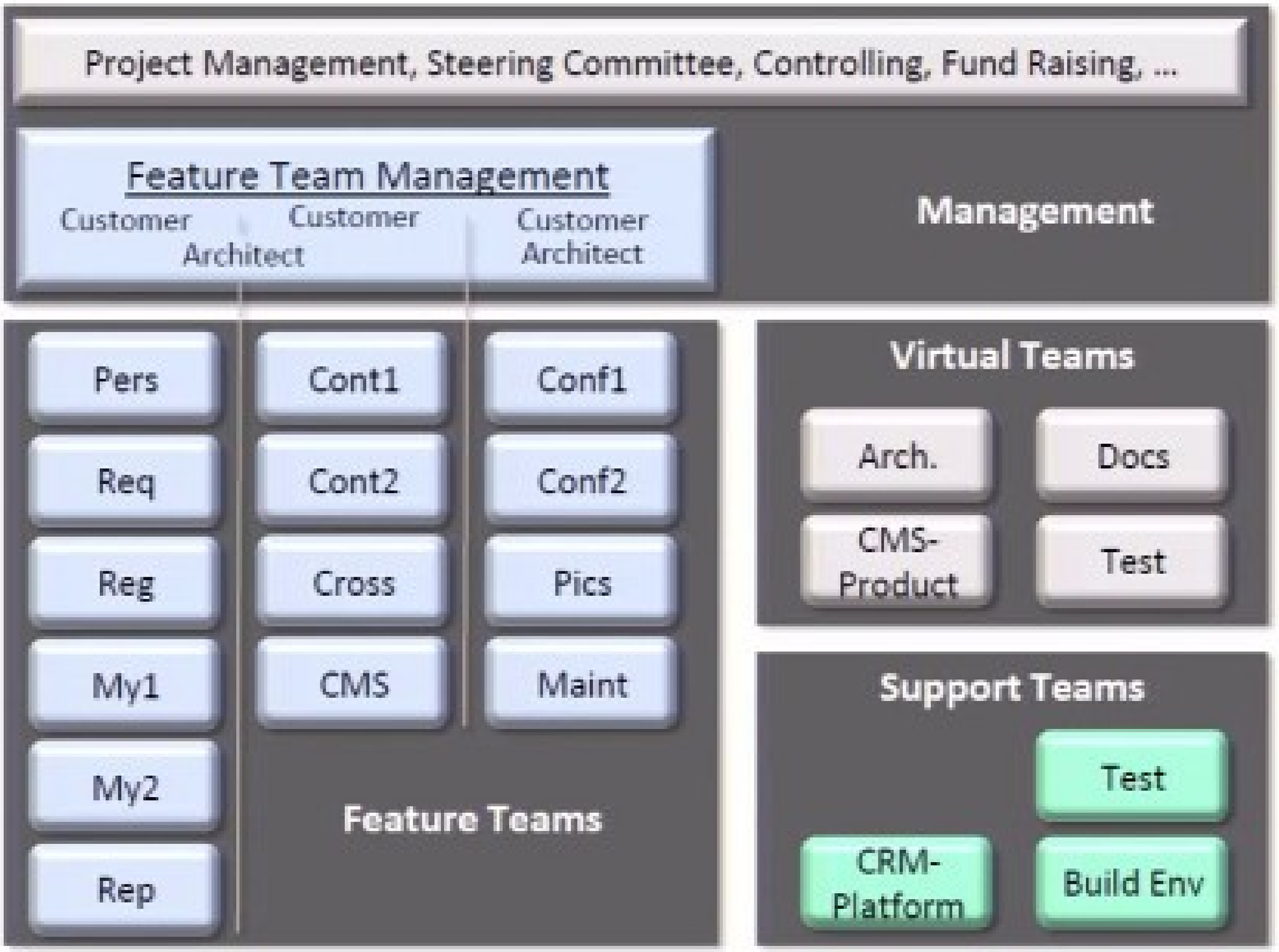
TNG TECHNOLOGY CONSULTING

# Organization Chart



TNG TECHNOLOGY CONSULTING

# Organization Chart



## Feature Teams

- Members
  - 7 +/- 2 developers (all external, many freelancers)
  - usually a user experience expert
  - PO (external, most from a marketing agency; worked very well)
  - SM (external, hired by SM experience, responsible for two or three teams; problematic)
- Goal: develop features
- Over time they became component teams to a certain degree
  → dependencies increased

- Remark:
  - for political reasons (e.g. different budgets) some feature teams were listed as support teams in the real project
  - in this case study I treat them as feature teams

## Feature Teams

- Members
  - 7 +/- 2 developers (all external, many freelancers)
  - usually a user experience expert
  - PO (external, most from a marketing agency; worked very well)
  - SM (external, hired by SM experience, responsible for two or three teams; problematic)
- Goal: develop features
- Over time they became component teams to a certain degree
  → dependencies increased

- Remark:
  - for political reasons (e.g. different budgets) some feature teams were listed as support teams in the real project
  - in this case study I treat them as feature teams

## Feature Teams

- Members
  - 7 +/- 2 developers (all external, many freelancers)
  - usually a user experience expert
  - PO (external, most from a marketing agency; worked very well)
  - SM (external, hired by SM experience, responsible for two or three teams; problematic)
- Goal: develop features
- Over time they became component teams to a certain degree
  → dependencies increased

- Remark:
  - for political reasons (e.g. different budgets) some feature teams were listed as support teams in the real project
  - in this case study I treat them as feature teams

TNG TECHNOLOGY CONSULTING

# Virtual Teams and Support Teams

- Virtual Teams
  - members
    - team lead
    - representatives from each feature team
    - permanent members → not really a Virtual Team; quite similar to support teams
  - goal
    - identify cross-team issues
    - work out possible solutions
    - define constraints* (architecture, documentation, test)

* Constraint = non-functional requirement

TNG TECHNOLOGY CONSULTING

# Virtual Teams and Support Teams

- Virtual Teams
  - members
    - team lead
    - representatives from each feature team
    - permanent members → not really a Virtual Team; quite similar to support teams
  - goal
    - identify cross-team issues
    - work out possible solutions
    - define constraints* (architecture, documentation, test)

* Constraint = non-functional requirement

TNG TECHNOLOGY CONSULTING

## Virtual Teams and Support Teams

- Virtual Teams
  - members
    - team lead
    - representatives from each feature team
    - permanent members → not really a Virtual Team; quite similar to support teams
  - goal
    - identify cross-team issues
    - work out possible solutions
    - define constraints* (architecture, documentation, test)

* Constraint = non-functional requirement

TNG TECHNOLOGY CONSULTING

## Virtual Teams and Support Teams

- Virtual Teams
  - members
    - team lead
    - representatives from each feature team
    - permanent members → not really a Virtual Team; quite similar to support teams
  - goal
    - identify cross-team issues
    - work out possible solutions
    - define constraints* (architecture, documentation, test)

- Support Teams
  - started with Scrum (PO, SM, sprints, planning, …)
  - drifted to different processes
    - Build Env Team: ticket process as usual in operations
    - Test Team: classical test management, manual testing, Gantt charts
  - problematic attitude: constraining the feature teams instead of supporting them

* Constraint = non-functional requirement

TNG TECHNOLOGY CONSULTING

## Virtual Teams and Support Teams

- Virtual Teams
  - members
    - team lead
    - representatives from each feature team
    - permanent members → not really a Virtual Team; quite similar to support teams
  - goal
    - identify cross-team issues
    - work out possible solutions
    - define constraints* (architecture, documentation, test)

- Support Teams
  - started with Scrum (PO, SM, sprints, planning, …)
  - drifted to different processes
    - Build Env Team: ticket process as usual in operations
    - Test Team: classical test management, manual testing, Gantt charts
  - problematic attitude: constraining the feature teams instead of supporting them

* Constraint = non-functional requirement

TNG TECHNOLOGY CONSULTING

# Virtual Teams and Support Teams

- Virtual Teams
  - members
    - team lead
    - representatives from each feature team
    - permanent members → not really a Virtual Team; quite similar to support teams
  - goal
    - identify cross-team issues
    - work out possible solutions
    - define constraints* (architecture, documentation, test)

- Support Teams
  - started with Scrum (PO, SM, sprints, planning, …)
  - drifted to different processes
    - Build Env Team: ticket process as usual in operations
    - Test Team: classical test management, manual testing, Gantt charts
  - problematic attitude: constraining the feature teams instead of supporting them

* Constraint = non-functional requirement

TNG TECHNOLOGY CONSULTING

# Virtual Teams and Support Teams

- Virtual Teams
  - members
    - team lead
    - representatives from each feature team
    - permanent members → not really a Virtual Team; quite similar to support teams
  - goal
    - identify cross-team issues
    - work out possible solutions
    - define constraints* (architecture, documentation, test)

- Support Teams
  - started with Scrum (PO, SM, sprints, planning, …)
  - drifted to different processes
    - Build Env Team: ticket process as usual in operations
    - Test Team: classical test management, manual testing, Gantt charts
  - problematic attitude: constraining the feature teams instead of supporting them

* Constraint = non-functional requirement

TNG TECHNOLOGY CONSULTING

# Virtual Teams and Support Teams

- Virtual Teams
  - members
    - team lead
    - representatives from each feature team
    - permanent members → not really a Virtual Team; quite similar to support teams
  - goal
    - identify cross-team issues
    - work out possible solutions
    - define constraints* (architecture, documentation, test)

- Support Teams
  - started with Scrum (PO, SM, sprints, planning, …)
  - drifted to different processes
    - Build Env Team: ticket process as usual in operations
    - Test Team: classical test management, manual testing, Gantt charts
  - problematic attitude: constraining the feature teams instead of supporting them

* Constraint = non-functional requirement

## Management

- Project Management:
  - budget
  - time
  - process
  - administration
  - Chief PO

- Feature Team Management (responsible for problems that the teams cannot solve by themselves):
  - joint prioritization of user stories, dependencies, constraints
  - set up new teams
  - dissolve teams, e.g.
    - PO left, no replacement → developers distributed to other teams
    - team too slow → buy product instead
  - restructure teams, e.g. temporary task force of experts from several teams

TNG TECHNOLOGY CONSULTING

## Management

- Project Management:
  - budget
  - time
  - process
  - administration
  - Chief PO

- Feature Team Management (responsible for problems that the teams cannot solve by themselves):
  - joint prioritization of user stories, dependencies, constraints
  - set up new teams
  - dissolve teams, e.g.
    - PO left, no replacement → developers distributed to other teams
    - team too slow → buy product instead
  - restructure teams, e.g. temporary task force of experts from several teams

TNG TECHNOLOGY CONSULTING

## Management

- Project Management:
  - budget
  - time
  - process
  - administration
  - Chief PO

- Feature Team Management (responsible for problems that the teams cannot solve by themselves):
  - joint prioritization of user stories, dependencies, constraints
  - set up new teams
  - dissolve teams, e.g.
    - PO left, no replacement → developers distributed to other teams
    - team too slow → buy product instead
  - restructure teams, e.g. temporary task force of experts from several teams

TNG TECHNOLOGY CONSULTING

## Management

- Project Management:
  - budget
  - time
  - process
  - administration
  - Chief PO

- Feature Team Management (responsible for problems that the teams cannot solve by themselves):
  - joint prioritization of user stories, dependencies, constraints
  - set up new teams
  - dissolve teams, e.g.
    - PO left, no replacement → developers distributed to other teams
    - team too slow → buy product instead
  - restructure teams, e.g. temporary task force of experts from several teams

# Agenda

- Theory
- Case Study
- Teams
- **Our Process**
- Challenges
- Lessons Learned

TNG TECHNOLOGY CONSULTING

## Process

- Scrum with two-week sprints, all feature teams in parallel
- Some support teams with an offset (infrastructure changes during planning day, moved to a different process anyway)
- During planning I + II: Scrum of Scrums (every two hours) to discuss dependencies
- Dailies (15 minutes every day)
  - Daily Scrum of Scrums
  - PO Daily
  - SM Daily
- Weeklies (one hour every week)
  - Virtual Team meeting
  - PO Weekly
  - SM Weekly
  - Feature Team Management Weekly

## Process

- PO approval during the sprint (mostly at the end of the sprint; stable software needed)
- Customer Review:
  - on the last day of the sprint
  - 30 minutes for each PO (one after the other)
  - presentation of finished user stories to Feature Team Management (Customer + Architect)
  - turned into a status report, instead of inspecting the developed features
- Joint Review
  - 1 – 2 hours in the evening after the customer review
  - short presentation of accepted user stories
  - whole project team (large room!)
- Retrospectives in addition to the team retrospectives
  - SMs: almost every sprint
  - POs: quarterly
  - Virtual Teams: quarterly

## Process

- PO approval during the sprint (mostly at the end of the sprint; stable software needed)
- Customer Review:
  - on the last day of the sprint
  - 30 minutes for each PO (one after the other)
  - presentation of finished user stories to Feature Team Management (Customer + Architect)
  - turned into a status report, instead of inspecting the developed features
- Joint Review
  - 1 – 2 hours in the evening after the customer review
  - short presentation of accepted user stories
  - whole project team (large room!)
- Retrospectives in addition to the team retrospectives
  - SMs: almost every sprint
  - POs: quarterly
  - Virtual Teams: quarterly

TNG TECHNOLOGY CONSULTING

## Process

- PO approval during the sprint (mostly at the end of the sprint; stable software needed)
- Customer Review:
  - on the last day of the sprint
  - 30 minutes for each PO (one after the other)
  - presentation of finished user stories to Feature Team Management (Customer + Architect)
  - turned into a status report, instead of inspecting the developed features
- Joint Review
  - 1 – 2 hours in the evening after the customer review
  - short presentation of accepted user stories
  - whole project team (large room!)
- Retrospectives in addition to the team retrospectives
  - SMs: almost every sprint
  - POs: quarterly
  - Virtual Teams: quarterly

## Process

- PO approval during the sprint (mostly at the end of the sprint; stable software needed)
- Customer Review:
  - on the last day of the sprint
  - 30 minutes for each PO (one after the other)
  - presentation of finished user stories to Feature Team Management (Customer + Architect)
  - turned into a status report, instead of inspecting the developed features
- Joint Review
  - 1 – 2 hours in the evening after the customer review
  - short presentation of accepted user stories
  - whole project team (large room!)
- Retrospectives in addition to the team retrospectives
  - SMs: almost every sprint
  - POs: quarterly
  - Virtual Teams: quarterly

TNG TECHNOLOGY CONSULTING

## Scrum of Scrums

- Daily Scrum of Scrums degraded into a dependency tracking meeting

- PO Daily
  - unclear goal; which questions should be answered?
  - What should their board look like?
    Status of the user stories in the current sprint (not helpful)

- SM Daily
  - even after one year and many experiments no satisfying solution
  - unclear goal
  - task board for impediments
  - problems with the board
    - very different priorities (e.g. rest room towels vs. unstable build environment)
    - different durations (many impediments not solvable within one day, not even one sprint)

## Scrum of Scrums

- Daily Scrum of Scrums degraded into a dependency tracking meeting

- PO Daily
  - unclear goal; which questions should be answered?
  - What should their board look like?
    Status of the user stories in the current sprint (not helpful)

- SM Daily
  - even after one year and many experiments no satisfying solution
  - unclear goal
  - task board for impediments
  - problems with the board
    - very different priorities (e.g. rest room towels vs. unstable build environment)
    - different durations (many impediments not solvable within one day, not even one sprint)

TNG TECHNOLOGY CONSULTING

## Scrum of Scrums

- Daily Scrum of Scrums degraded into a dependency tracking meeting

- PO Daily
  - unclear goal; which questions should be answered?
  - What should their board look like?
    Status of the user stories in the current sprint (not helpful)

- SM Daily
  - even after one year and many experiments no satisfying solution
  - unclear goal
  - task board for impediments
  - problems with the board
    - very different priorities (e.g. rest room towels vs. unstable build environment)
    - different durations (many impediments not solvable within one day, not even one sprint)

## Scrum of Scrums

- Daily Scrum of Scrums degraded into a dependency tracking meeting

- PO Daily
  - unclear goal; which questions should be answered?
  - What should their board look like?
    Status of the user stories in the current sprint (not helpful)

- SM Daily
  - even after one year and many experiments no satisfying solution
  - unclear goal
  - task board for impediments
  - problems with the board
    - very different priorities (e.g. rest room towels vs. unstable build environment)
    - different durations (many impediments not solvable within one day, not even one sprint)

TNG TECHNOLOGY CONSULTING

## Dependencies

- Sometimes teams need changes in components they do not know sufficiently

- Dependency process
  - identification of dependencies in the planning meeting or during the sprint
  - written on sticky notes and brought to the Scrum of Scrums
  - huge matrix on the wall with a column and a row for each team
  - one copy for the dependency matrix, one copy for the supporting team
  - highest priority on the task board of the supporting team

- Problems:
  - easy to transfer work to a different team → many dependencies
  - agreed delivery dates for dependencies were often not met
  - deliveries often did not fit the needs → rework
  - team commitment obsolete
    - unplanned dependencies → no time for user stories
    - undelivered dependencies → dependent user stories fail

TNG TECHNOLOGY CONSULTING

## Dependencies

- Sometimes teams need changes in components they do not know sufficiently

- Dependency process
  - identification of dependencies in the planning meeting or during the sprint
  - written on sticky notes and brought to the Scrum of Scrums
  - huge matrix on the wall with a column and a row for each team
  - one copy for the dependency matrix, one copy for the supporting team
  - highest priority on the task board of the supporting team

- Problems:
  - easy to transfer work to a different team → many dependencies
  - agreed delivery dates for dependencies were often not met
  - deliveries often did not fit the needs → rework
  - team commitment obsolete
    - unplanned dependencies → no time for user stories
    - undelivered dependencies → dependent user stories fail

TNG TECHNOLOGY CONSULTING

## Dependencies

- Sometimes teams need changes in components they do not know sufficiently

- Dependency process
  - identification of dependencies in the planning meeting or during the sprint
  - written on sticky notes and brought to the Scrum of Scrums
  - huge matrix on the wall with a column and a row for each team
  - one copy for the dependency matrix, one copy for the supporting team
  - highest priority on the task board of the supporting team

- Problems:
  - easy to transfer work to a different team → many dependencies
  - agreed delivery dates for dependencies were often not met
  - deliveries often did not fit the needs → rework
  - team commitment obsolete
    - unplanned dependencies → no time for user stories
    - undelivered dependencies → dependent user stories fail

TNG TECHNOLOGY CONSULTING

## Prerequisites

- Identify dependencies earlier: "prerequisites"

- Prerequisite process
  - show up when writing user stories
  - the PO talks to the other PO about the prerequisite
  - the PO of the supporting team writes a so called prerequisite user story
  - it is put into the backlog with highest priority

- Facilitation
  - PO meeting on a backlog board to identify prerequisites
  - worked "shockingly well"

- Problems:
  - more prerequisites than user stories in the backlog of some teams
  - even more waste: coordination, waiting, non-fitting deliveries, rework

TNG TECHNOLOGY CONSULTING

## Prerequisites

- Identify dependencies earlier: "prerequisites"

- Prerequisite process
  - show up when writing user stories
  - the PO talks to the other PO about the prerequisite
  - the PO of the supporting team writes a so called prerequisite user story
  - it is put into the backlog with highest priority

- Facilitation
  - PO meeting on a backlog board to identify prerequisites
  - worked "shockingly well"

- Problems:
  - more prerequisites than user stories in the backlog of some teams
  - even more waste: coordination, waiting, non-fitting deliveries, rework

TNG TECHNOLOGY CONSULTING

## Prerequisites

- Identify dependencies earlier: "prerequisites"

- Prerequisite process
  - show up when writing user stories
  - the PO talks to the other PO about the prerequisite
  - the PO of the supporting team writes a so called prerequisite user story
  - it is put into the backlog with highest priority

- Facilitation
  - PO meeting on a backlog board to identify prerequisites
  - worked "shockingly well"

- Problems:
  - more prerequisites than user stories in the backlog of some teams
  - even more waste: coordination, waiting, non-fitting deliveries, rework

TNG TECHNOLOGY CONSULTING

## Prerequisites

- Identify dependencies earlier: "prerequisites"

- Prerequisite process
  - show up when writing user stories
  - the PO talks to the other PO about the prerequisite
  - the PO of the supporting team writes a so called prerequisite user story
  - it is put into the backlog with highest priority

- Facilitation
  - PO meeting on a backlog board to identify prerequisites
  - worked "shockingly well"

- Problems:
  - more prerequisites than user stories in the backlog of some teams
  - even more waste: coordination, waiting, non-fitting deliveries, rework

TNG TECHNOLOGY CONSULTING

## Version Control and Deployment Process

- Version Control
  - whole project on the same SVN repository
  - strategy: unstable trunk
  - stabilization branch on the last two days of the sprint
- Staging Environment
  - DEV server: hourly deployment
  - TEST server: nightly deployment
  - INT server: deployment at the end of the sprint
- Weaknesses
  - one check-in can block the whole project
  - almost no quality gate for check-ins
  - little familiarity with agile software development practices (TDD, feature toggles)
  - time triggered deployments
- Result
  - dysfunctional software for several hours, days, sometimes even weeks
  - time-consuming stabilization phase every sprint

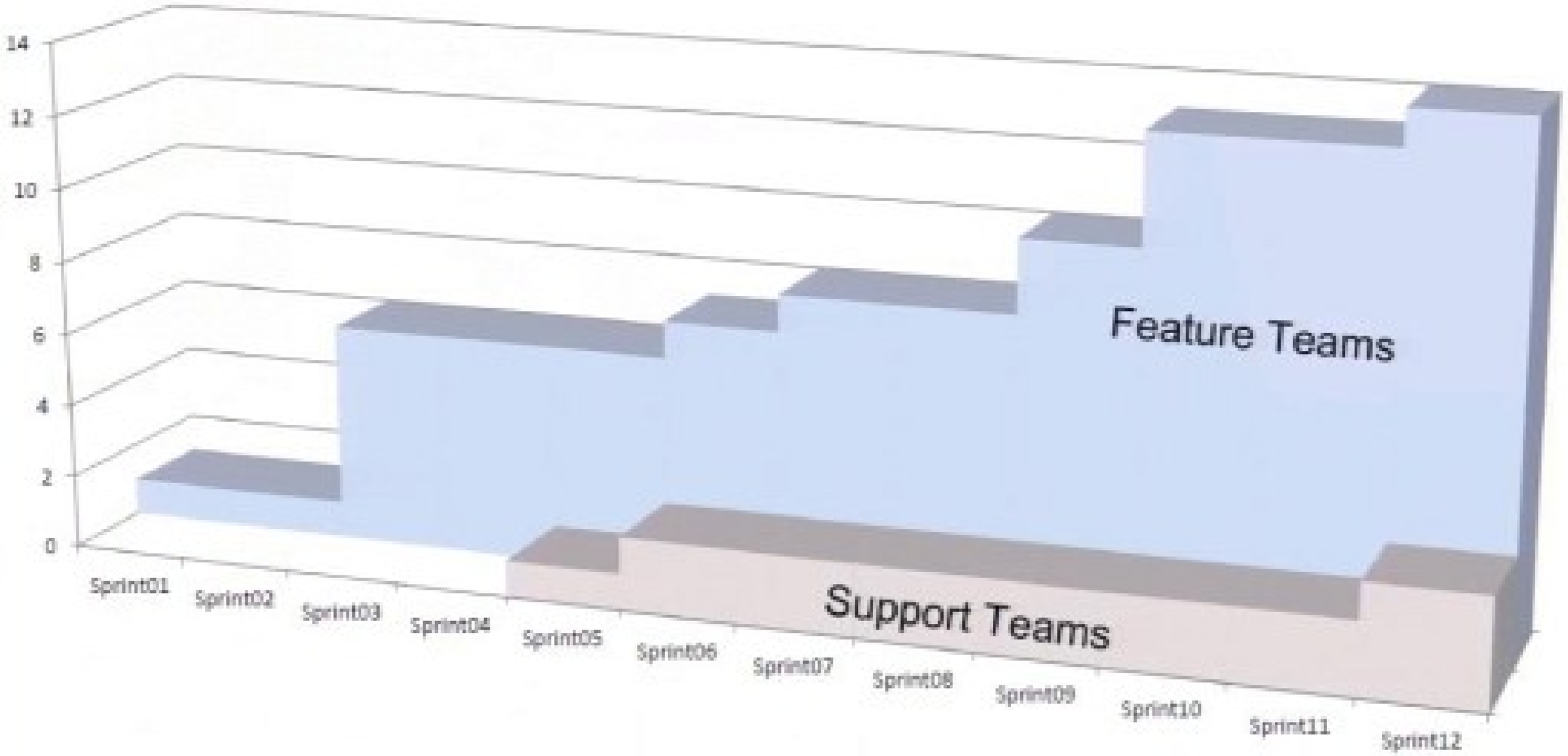
TNG TECHNOLOGY CONSULTING

# Agenda

- Theory
- Case Study
- Teams
- Our Process
- **Challenges**
- Lessons Learned

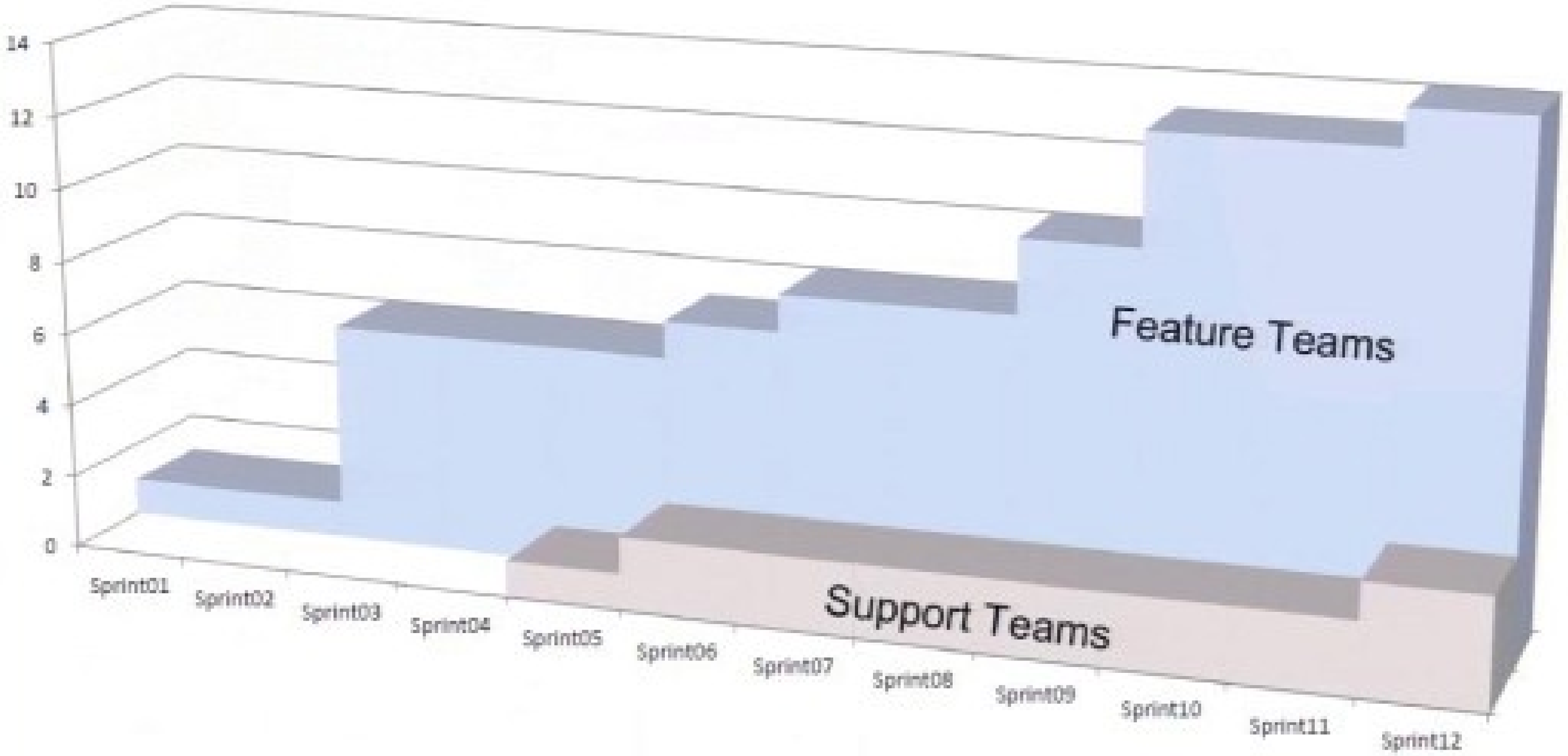
TNG TECHNOLOGY CONSULTING

# Team Scaling



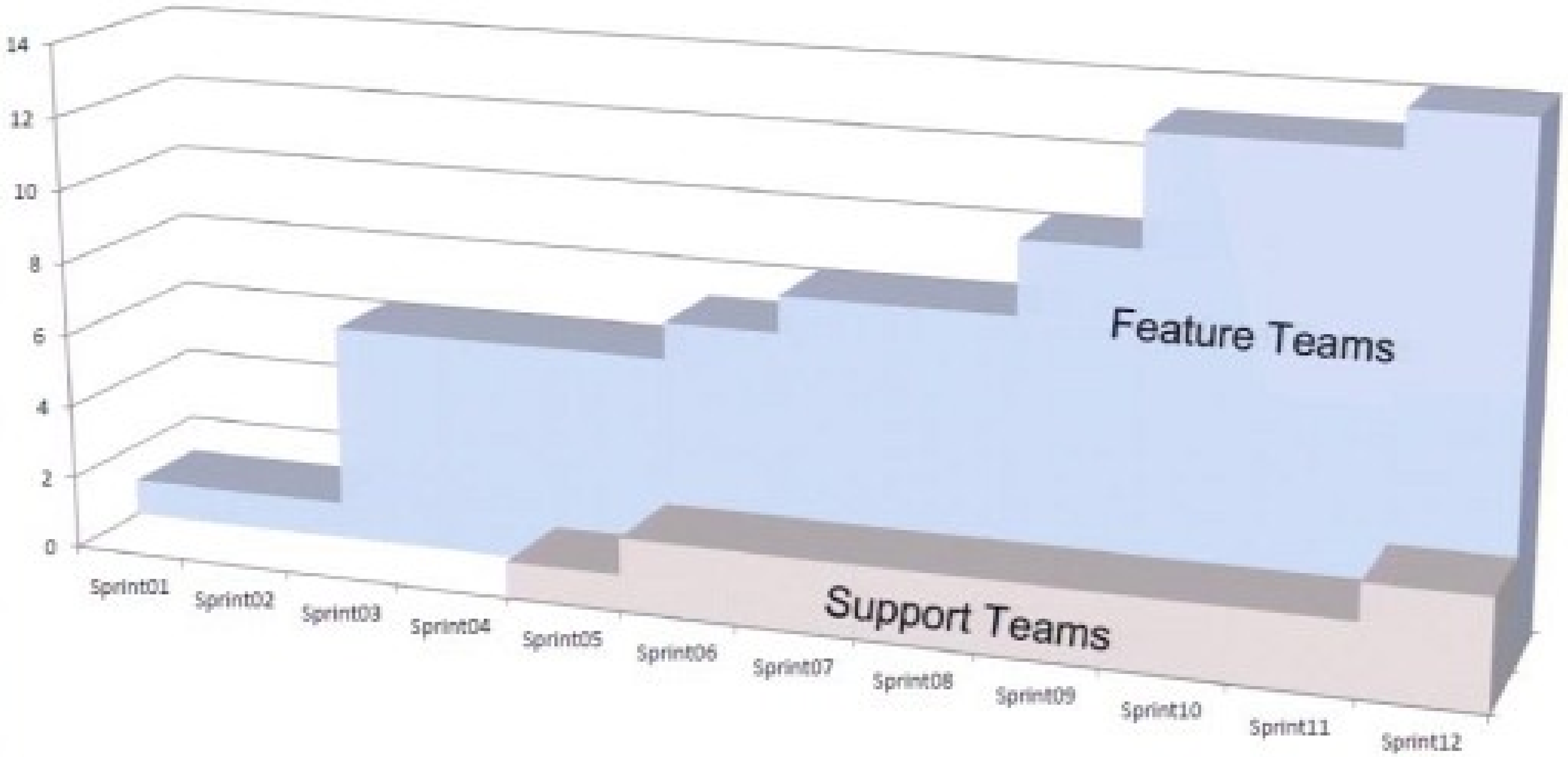
TNG TECHNOLOGY CONSULTING

# Team Scaling



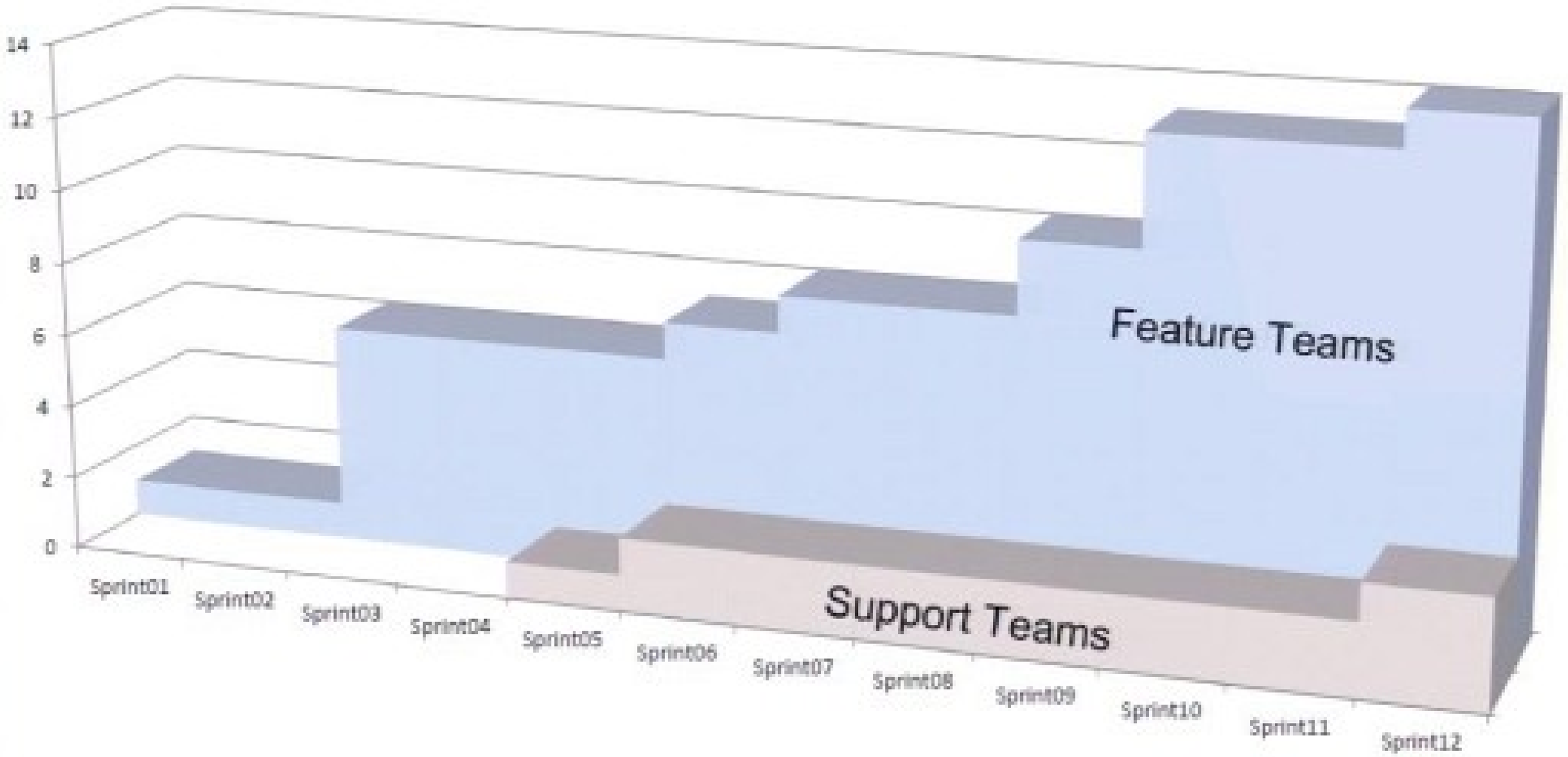
TNG TECHNOLOGY CONSULTING

# Team Scaling



TNG TECHNOLOGY CONSULTING

# Team Scaling



TNG TECHNOLOGY CONSULTING

## Problems with Scaling

- Teams were set up in so called "waves"

- First wave of five teams rolled in
  - broken builds
  - disabled unit tests
  - not enough space in the office

- Next waves came too fast
  - Every time we started getting the infrastructure and software working, the next wave broke everything again

TNG TECHNOLOGY CONSULTING

# Problems with Long Term Planning

- Management could not find out how to use the story points for their planning
  - not comparable for different teams
  - no trust in story points

- Hence fall-back to classical methods
  - "experts" estimated person days
  - releases with fixed scope were defined
  - six weeks test phase was planned
  - with four weeks buffer in addition

- Consequences
  - Definition of Done was softened
  - features presented although not yet done
  - teams built barriers, dependencies increased
  - many defects

TNG TECHNOLOGY CONSULTING

## Problems with Long Term Planning

- Management could not find out how to use the story points for their planning
  - not comparable for different teams
  - no trust in story points

- Hence fall-back to classical methods
  - "experts" estimated person days
  - releases with fixed scope were defined
  - six weeks test phase was planned
  - with four weeks buffer in addition

- Consequences
  - Definition of Done was softened
  - features presented although not yet done
  - teams built barriers, dependencies increased
  - many defects

TNG TECHNOLOGY CONSULTING

## Problems with Long Term Planning

- Management could not find out how to use the story points for their planning
  - not comparable for different teams
  - no trust in story points

- Hence fall-back to classical methods
  - "experts" estimated person days
  - releases with fixed scope were defined
  - six weeks test phase was planned
  - with four weeks buffer in addition

- Consequences
  - Definition of Done was softened
  - features presented although not yet done
  - teams built barriers, dependencies increased
  - many defects

TNG TECHNOLOGY CONSULTING

# Launch in the First Market

- Finally we got the problems under control and went live in the first market

- Success:
  - almost no production defects
  - the other markets want us
  - offer money to get it earlier

- Steering committee member at the launch party:

  "Without Scrum we would still write documents."

- The lessons learned influenced the next phase

# Agenda

- Theory
- Case Study
- Teams
- Our Process
- Challenges
- **Lessons Learned**

TNG TECHNOLOGY CONSULTING

## Current Development for Further Markets

- Project management:
  - "agile" is the right way
  - improve the organization in this direction

- Explicit areas with an area PO and a lead developer
- No dependency process anymore; instead responsibility for complete feature in one team and collaboration
- No Scrum of Scrums anymore
- No official Virtual Teams anymore
  - instead more support teams
  - weekly meeting with team representatives
- Long term planning still by experts; however, feedback from the teams is gathered:
  - estimates were compared with actual costs
  - story points were scaled to person days using the velocity → comparable across teams
- Testers in every team; more test automation
- Scrum Master role embodied by one developer per team (part time)

TNG TECHNOLOGY CONSULTING

## Current Development for Further Markets

- Project management:
  - "agile" is the right way
  - improve the organization in this direction

- Explicit areas with an area PO and a lead developer
- No dependency process anymore; instead responsibility for complete feature in one team and collaboration
- No Scrum of Scrums anymore
- No official Virtual Teams anymore
  - instead more support teams
  - weekly meeting with team representatives
- Long term planning still by experts; however, feedback from the teams is gathered:
  - estimates were compared with actual costs
  - story points were scaled to person days using the velocity → comparable across teams
- Testers in every team; more test automation
- Scrum Master role embodied by one developer per team (part time)

TNG TECHNOLOGY CONSULTING

## Current Development for Further Markets

- Project management:
  - "agile" is the right way
  - improve the organization in this direction

- Explicit areas with an area PO and a lead developer
- No dependency process anymore; instead responsibility for complete feature in one team and collaboration
- No Scrum of Scrums anymore
- No official Virtual Teams anymore
  - instead more support teams
  - weekly meeting with team representatives
- Long term planning still by experts; however, feedback from the teams is gathered:
  - estimates were compared with actual costs
  - story points were scaled to person days using the velocity → comparable across teams
- Testers in every team; more test automation
- Scrum Master role embodied by one developer per team (part time)

TNG TECHNOLOGY CONSULTING

## Current Development for Further Markets

- Project management:
  - "agile" is the right way
  - improve the organization in this direction

- Explicit areas with an area PO and a lead developer
- No dependency process anymore; instead responsibility for complete feature in one team and collaboration
- No Scrum of Scrums anymore
- No official Virtual Teams anymore
  - instead more support teams
  - weekly meeting with team representatives
- Long term planning still by experts; however, feedback from the teams is gathered:
  - estimates were compared with actual costs
  - story points were scaled to person days using the velocity → comparable across teams
- Testers in every team; more test automation
- Scrum Master role embodied by one developer per team (part time)

## Conclusion

- Daily Scrum of Scrums is not missed
- PO Daily survived self-organized
- Dependencies:
  - much better now
  - some misunderstandings about collective code ownership
- Areas:
  - too many small areas; area PO or team PO superfluous in some areas
  - lead developer: to be seen
- Support teams: service orientation unclear; potentially wrong direction:
  - in the upper floor, same as project management
  - different name: "synchro teams", because they synchronize the feature teams
- Long term planning: to be seen after the next launch
- Testers in the teams: seems to be good; however they are not integrated optimally yet
- Scrum Masters from the teams: works very well
- Version control and deployment: still room for improvement

## Conclusion

- Daily Scrum of Scrums is not missed
- PO Daily survived self-organized
- Dependencies:
  - much better now
  - some misunderstandings about collective code ownership
- Areas:
  - too many small areas; area PO or team PO superfluous in some areas
  - lead developer: to be seen
- Support teams: service orientation unclear; potentially wrong direction:
  - in the upper floor, same as project management
  - different name: "synchro teams", because they synchronize the feature teams
- Long term planning: to be seen after the next launch
- Testers in the teams: seems to be good; however they are not integrated optimally yet
- Scrum Masters from the teams: works very well
- Version control and deployment: still room for improvement

## Conclusion

- Daily Scrum of Scrums is not missed
- PO Daily survived self-organized
- Dependencies:
  - much better now
  - some misunderstandings about collective code ownership
- Areas:
  - too many small areas; area PO or team PO superfluous in some areas
  - lead developer: to be seen
- Support teams: service orientation unclear; potentially wrong direction:
  - in the upper floor, same as project management
  - different name: "synchro teams", because they synchronize the feature teams
- Long term planning: to be seen after the next launch
- Testers in the teams: seems to be good; however they are not integrated optimally yet
- Scrum Masters from the teams: works very well
- Version control and deployment: still room for improvement

## Conclusion

- Daily Scrum of Scrums is not missed
- PO Daily survived self-organized
- Dependencies:
  - much better now
  - some misunderstandings about collective code ownership
- Areas:
  - too many small areas; area PO or team PO superfluous in some areas
  - lead developer: to be seen
- Support teams: service orientation unclear; potentially wrong direction:
  - in the upper floor, same as project management
  - different name: "synchro teams", because they synchronize the feature teams
- Long term planning: to be seen after the next launch
- Testers in the teams: seems to be good; however they are not integrated optimally yet
- Scrum Masters from the teams: works very well
- Version control and deployment: still room for improvement

## Conclusion

- Daily Scrum of Scrums is not missed
- PO Daily survived self-organized
- Dependencies:
  - much better now
  - some misunderstandings about collective code ownership
- Areas:
  - too many small areas; area PO or team PO superfluous in some areas
  - lead developer: to be seen
- Support teams: service orientation unclear; potentially wrong direction:
  - in the upper floor, same as project management
  - different name: "synchro teams", because they synchronize the feature teams
- Long term planning: to be seen after the next launch
- Testers in the teams: seems to be good; however they are not integrated optimally yet
- Scrum Masters from the teams: works very well
- Version control and deployment: still room for improvement

TNG TECHNOLOGY CONSULTING

# Thank you for your attention!


- Dr. Sebastian Stamminger
  sebastian.stamminger@tngtech.com
  Tel. +49 176 2191 5655