

# 5 “enemies” of teamwork in SAFe and how to fight them

Svetlana Bolsunovskaia  
PM, Motorola Solutions



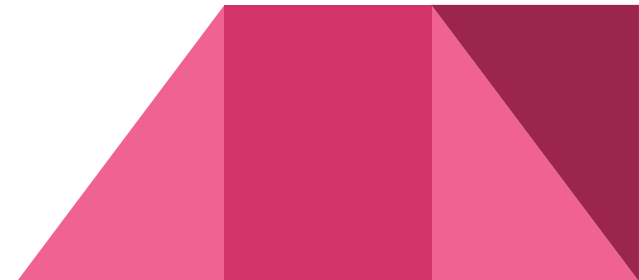
# Agenda

- ❖ Introduction
- ❖ SAFe Teamwork “enemies”
- ❖ On a way to team effectiveness
- ❖ Summary



# Introduction

What is  
Scaled **Agile** Framework (SAFe)?



# Agile Frameworks: SCRUM or SAFe ?

- ❑ Product execution

- ❑ **1,2 teams**

- ❑ **10-15 people in total**

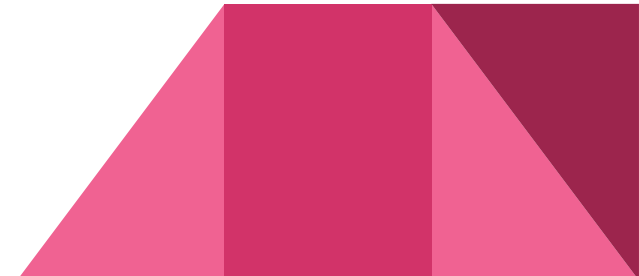
- ❑ Direct contact “team - customer”

- ❑ Product portfolio execution

- ❑ **3,4,5 teams** working on common products

- ❑ **30-50 people in total**

- ❑ No direct contact “team - customer”





# Project Description

5 development teams, 1  
testing team

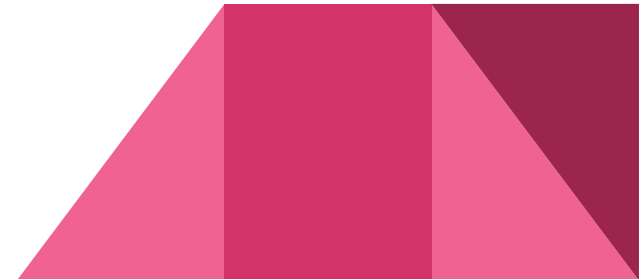
Architecture, Business  
teams

SAFe

Project Management  
Office (PMO)

4 locations

50 people



# 5 “enemies” of a teamwork



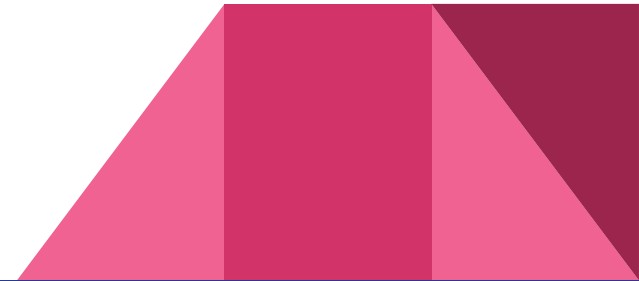
# Enemies

1. Unclear responsible areas
2. Not agreed goals
3. Internal competition
4. Not agreed common rules
5. Restrictions with face to face communication

---

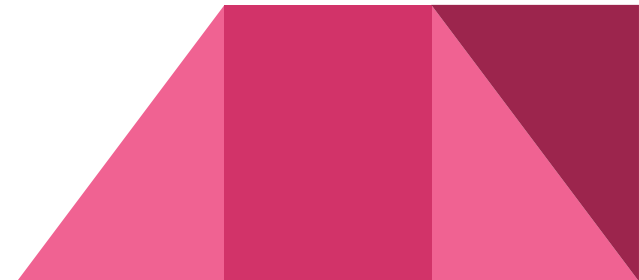
# Unclear responsible areas

- ❖ No owners for particular actions
- ❖ Not enough information sharing, as unclear who may be impacted



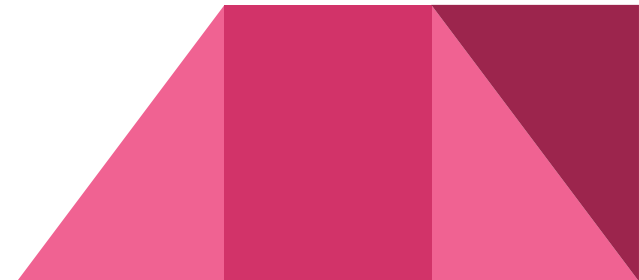
## Not agreed goals

- ❖ No clear team's understanding of project vision, project priorities
- ❖ Double efforts due to duplicating goals



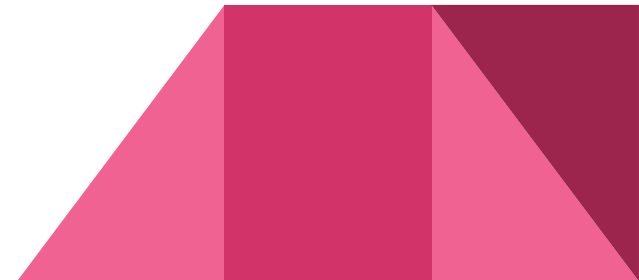
# Internal competition

- ❖ Teams do not share real work execution progress
- ❖ Imbalance between team's desire to look "good" and necessity to make high quality product



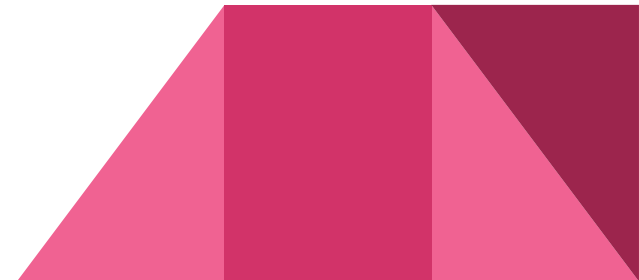
## Not agreed common rules

- ❖ Much time spent on program planning if no agreed format
- ❖ Unsynchronized team practises lead to additional communication efforts



# Restrictions with face to face communication

- ❖ Difficult to find contact person for specific question quickly
- ❖ Uncoordinated work as too much efforts to organize daily sync up with all teams





On a way to team effectiveness

# 3 Major steps to achieve effective teamwork

**Step 1:**  
Agree on responsibilities

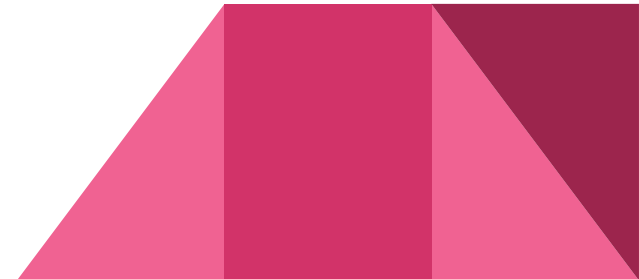
**Figure out, describe and explain responsibilities for key SAFe model roles**

**Step 2 :**  
Make consistent goals

**Identify and agree on goals for each SAFe model level - Portfolio, Program, Team**

**Step 3:**  
Synchronize teams work rules

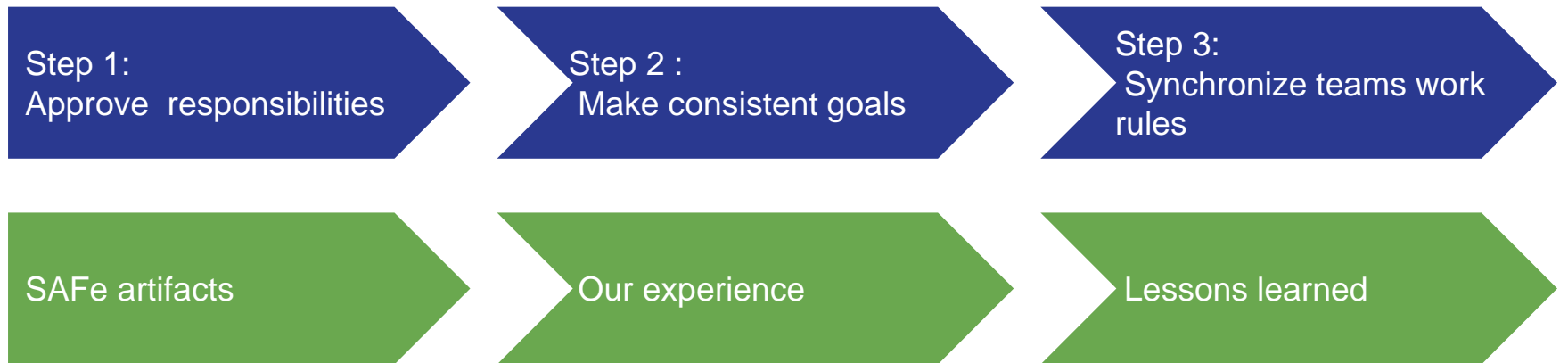
**Establish and document common rules and make easy to follow and clear to everybody**





# Practices to achieve teamwork effectiveness:

## 3 main aspects



# Step 1

## Agree on responsibilities

---

Responsibilities

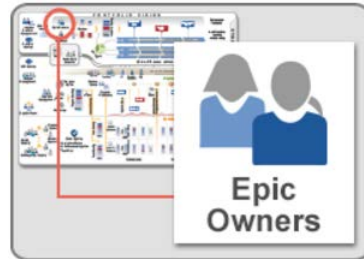
Common goals

Common rules

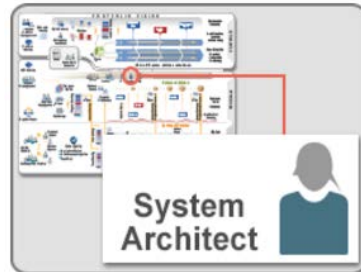
SAFe artifacts

Our experience

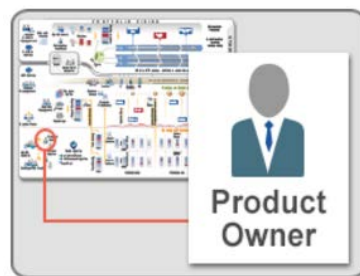
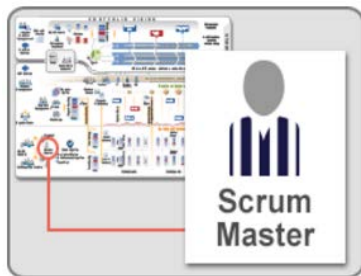
Lessons Learned



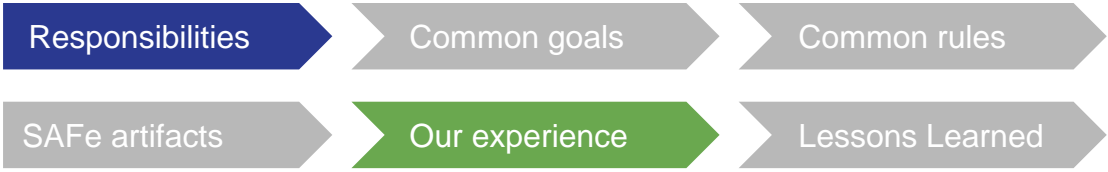
Portfolio



Program



Team

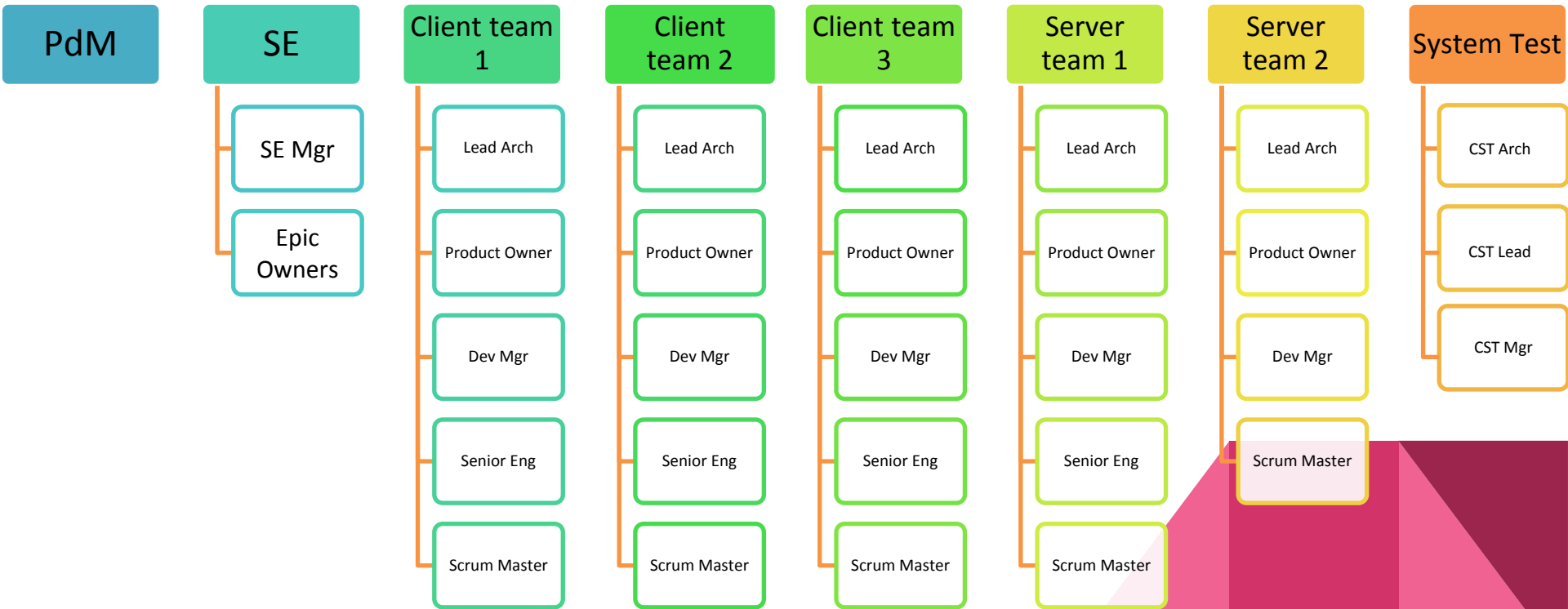


**Chief System Architect**

**Release Train Engineer (RTE)**

**Server Manager**

**Client Manager**



Responsibilities

Common goals

Common rules

SAFe artifacts

Our experience

Lessons Learned



Make sure that roles do not have overlapping responsibility (e.g. Epic owner, Product owner)



Avoid additional manager roles which are not included in the original SAFe process (e.g. Development manager, Client/ Server manager)

# Step 2

## Make consistent goals

---

Responsibilities

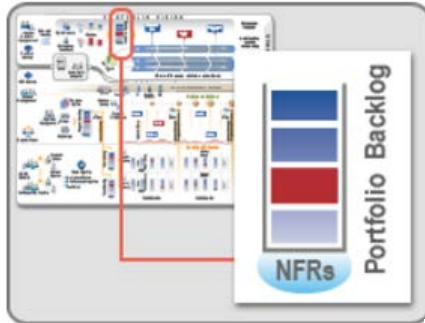
Common goals

Common rules

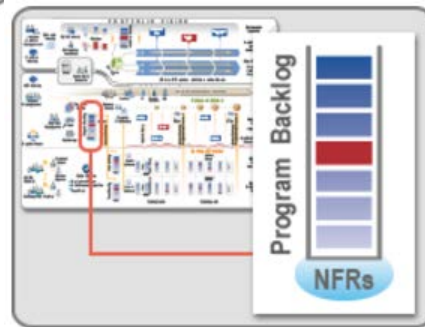
SAFe artifacts

Our experience

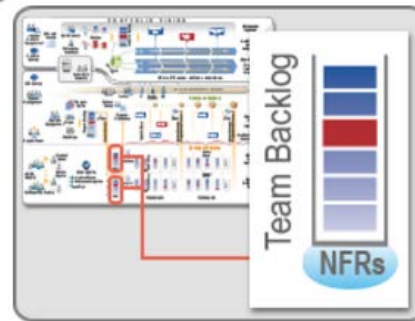
Lessons Learned



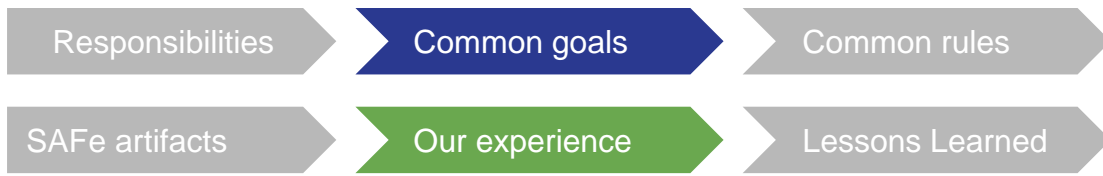
Portfolio Backlog



Program Backlog



Team Backlog



# Goals agreement process

Feature priorities

Epic priorities and plan

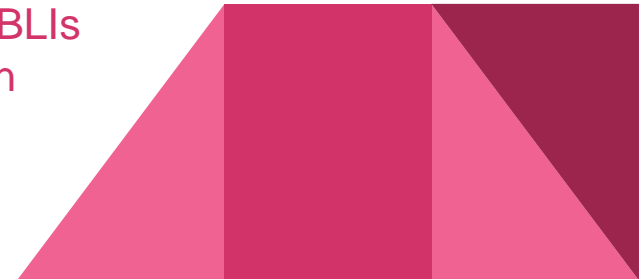
Backlog Items(BLIs) priorities and plan



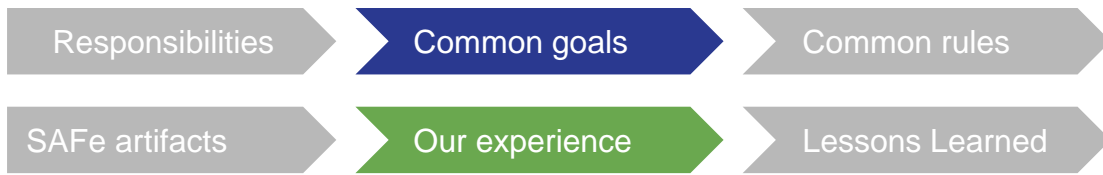
Goals for features delivery

Goals for Epics completion

Goals for BLIs completion







# Example of planning sessions schedule and artifacts

## Week 1

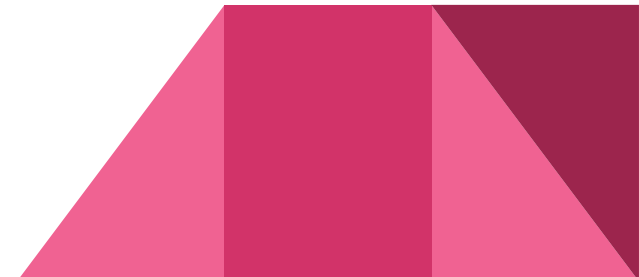
- Feature Owners present/review EPICs with Architects and Sub-teams
- Sub-team collaboration on EPICs and User Story (BLI) draft creation

## Week 2

- Sub-team collaboration on EPICs and User Stories (BLIs) finalization
- Make Estimates for User Stories (BLIs)

## Week 3

- Build EPIC completion and System Test plan
- Program Increment objective commitment



Responsibilities

Common goals

Common rules

SAFe artifacts

Our experience

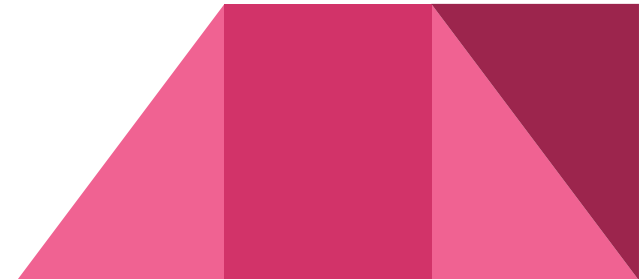
Lessons Learned



Avoid intra Epic dependencies  
between teams



Make Product Increment  
objective in accordance with  
Epics priorities



# **Step 3**

## Synchronize teams work rules

---

Responsibilities

Common goals

Common rules

SAFe artifacts

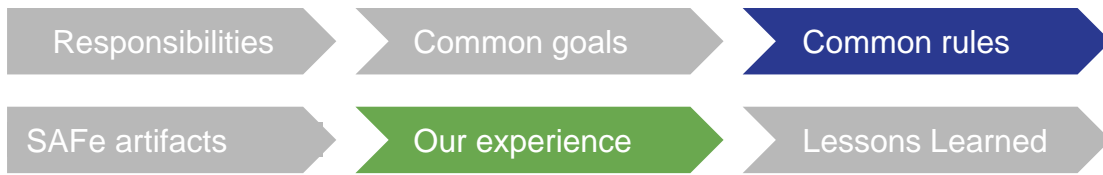
Our experience

Lessons Learned

Planning

Execution

Delivery



### Planning. *Practices*

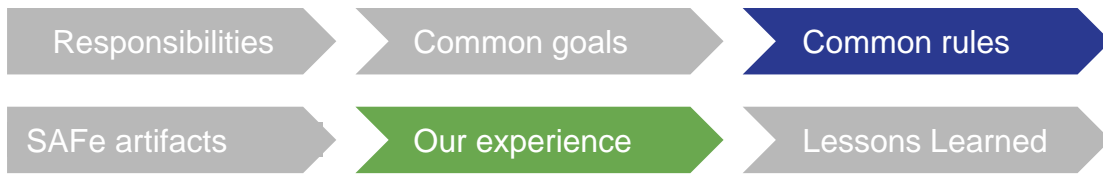
- One place for Program Increment planning session
- 2 weeks lag between PIs is required for valuable planning

### Execution. *Practices*

- Regular meeting to sync up on program status (e.g. weekly)
- Regular meeting to sync up on development status (e.g. weekly)

### Delivery. *Practices*

- Team's sprint review in addition to sprint demo
- Common demo for all teams for critical functionality



### Planning. *Agree on:*

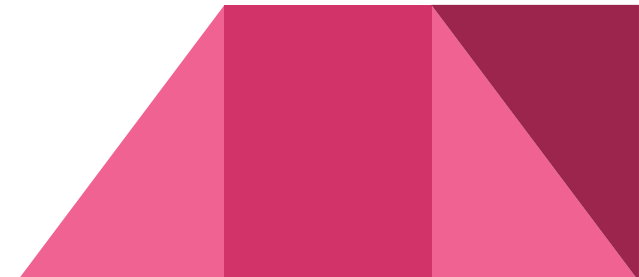
- ✓ Test strategy
- ✓ Quality goals, metrics
- ✓ Definition of Done

### Execution. *Agree on:*

- ✓ Rules related to common code
- ✓ Tools, equipment, environment

### Delivery. *Agree on:*

- ✓ Exact delivery time
- ✓ Required documentation
- ✓ Acceptance criteria



Responsibilities

Common goals

Common rules

SAFe artifacts

Our experience

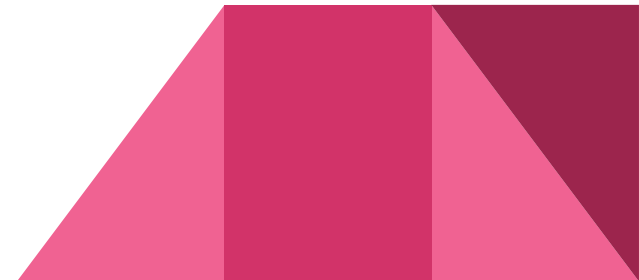
Lessons Learned



Make team's progress visible



Create blocking issues  
management process



# Summary

- ❖ SAFe overview **presented**
- ❖ Main teamwork “enemies” **disclosed**
- ❖ Established practices allowed to achieve teamwork effectiveness **shared**

Responsibilities

Common goals

Common rules

SAFe artifacts

Our experience

Lessons Learned





Thank you