#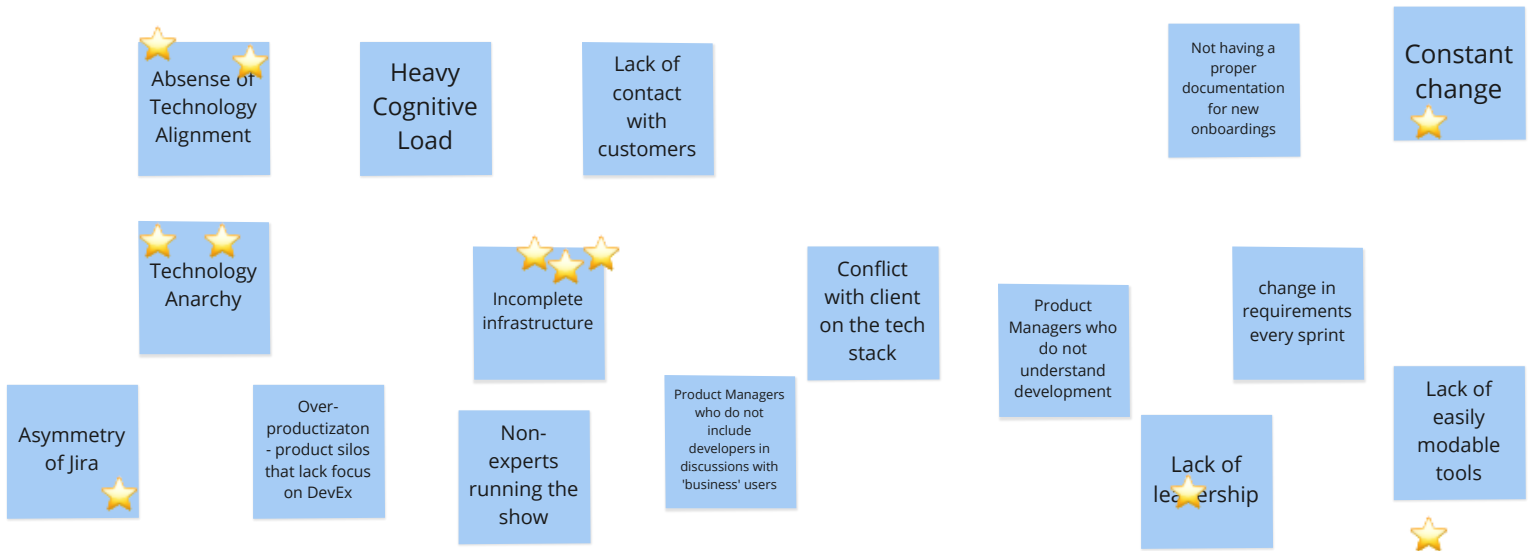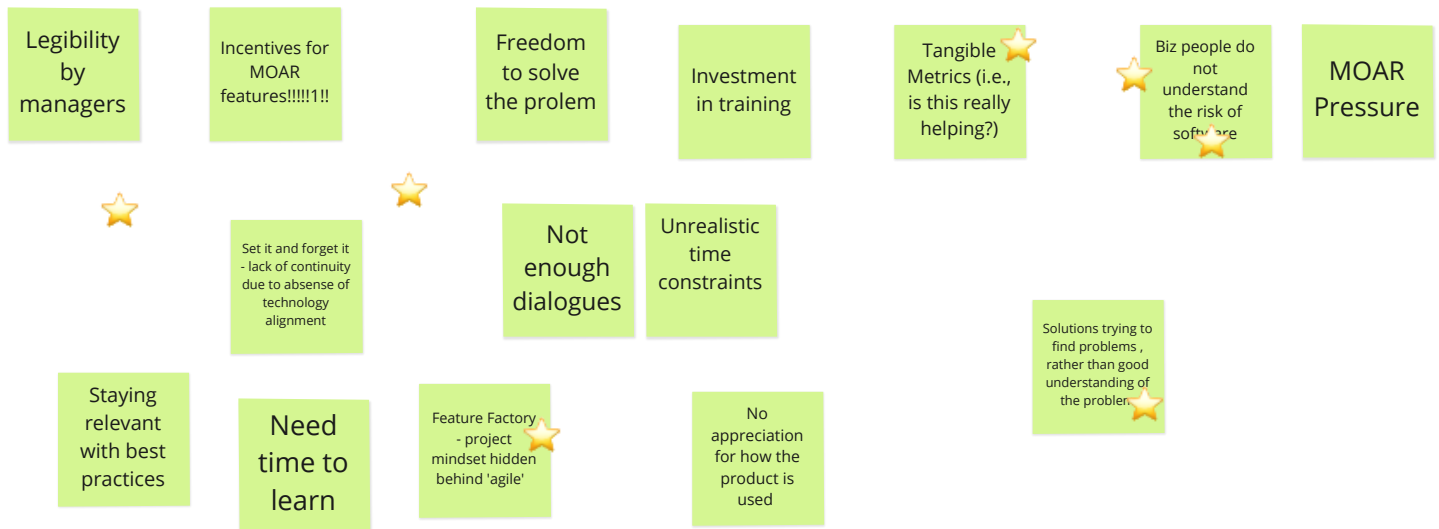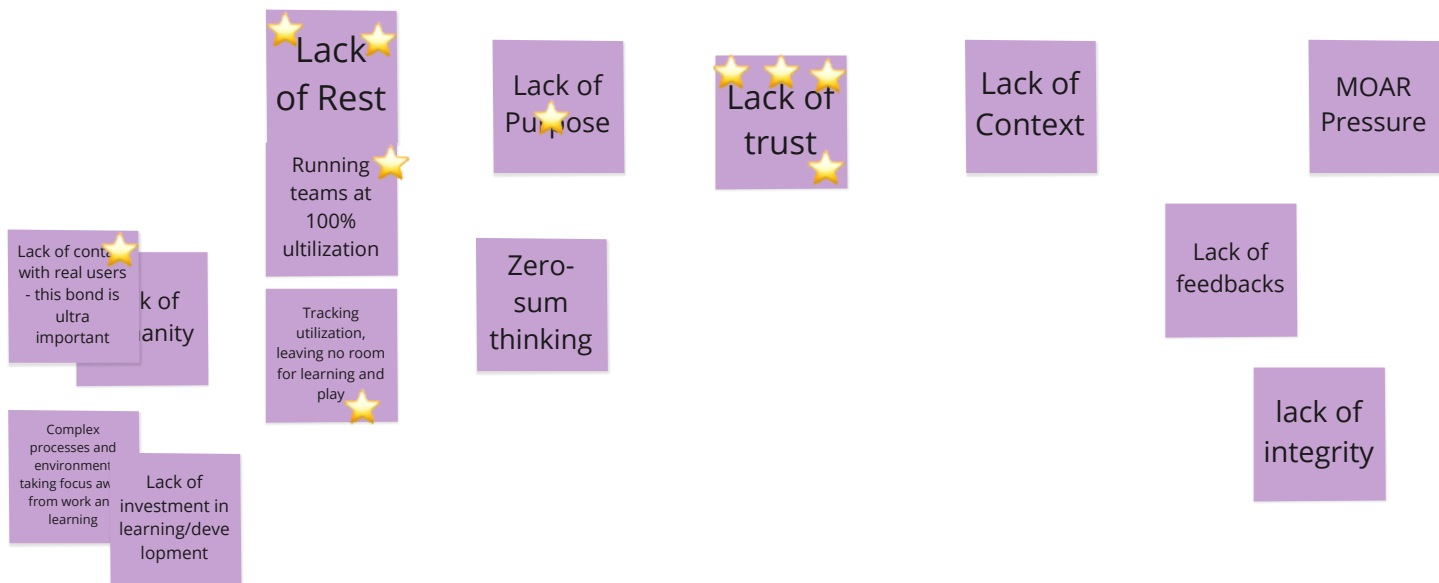 Using the lens of "**Developer Experience**", think about what problems and constraints make creating software difficult

- Absense of Technology Alignment ⭐
- Heavy Cognitive Load
- Lack of contact with customers
- Not having a proper documentation for new onboardings
- Constant change ⭐
- Technology Anarchy ⭐⭐
- Incomplete infrastructure ⭐⭐⭐
- Conflict with client on the tech stack
- Product Managers who do not understand development
- change in requirements every sprint
- Asymmetry of Jira ⭐
- Over-productizaton - product silos that lack focus on DevEx
- Non-experts running the show
- Product Managers who do not include developers in discussions with 'business' users
- Lack of leadership ⭐
- Lack of easily modable tools ⭐

# Using the lens of "**Software Excellence**", think about what problems and constraints make it difficult to build a product that matters, and do awesome quality work:

- Legibility by managers
- Incentives for MOAR features!!!!!1!!
- Freedom to solve the prolem
- Investment in training
- Tangible Metrics (i.e., is this really helping?) ⭐
- Biz people do not understand the risk of software ⭐⭐
- MOAR Pressure
- ⭐
- ⭐
- Set it and forget it - lack of continuity due to absense of technology alignment
- Not enough dialogues
- Unrealistic time constraints
- Solutions trying to find problems , rather than good understanding of the problem ⭐
- Staying relevant with best practices
- Need time to learn
- Feature Factory - project mindset hidden behind 'agile' ⭐
- No appreciation for how the product is used

# Using the lens of "**Thriving**", think about what problems and constraints make it difficult for people to thrive, teams to thrive, and organizations to thrive:

- Lack of Rest ⭐⭐⭐
- Lack of Purpose ⭐
- Lack of trust ⭐⭐⭐⭐
- Lack of Context
- MOAR Pressure
- Running teams at 100% ultilization ⭐
- Lack of contact with real users - this bond is ultra important ⭐
- ...k of ...anity
- Tracking utilization, leaving no room for learning and play ⭐
- Zero-sum thinking
- Lack of feedbacks
- lack of integrity
- Complex processes and environment taking focus aw... from work an... learning
- Lack of investment in learning/development

Copy the Stars!

⭐⭐
⭐

Incomplete infrastructure

Too much focus on "creating software", not enough on "solving problems"

Absense of Technology Alignment

Constant change

Lack of easily modable tools

Lack of leadership

Incomplete infrastructure

Technology Anarchy

Asymmetry of Jira

Biz people do not understand the risk of software

Biz people do not understand the risk of software

trendy technology stack choices

Too much focus on "creating software", not enough on "solving problems"
⭐⭐⭐ ⭐ ⭐ ⭐ ⭐

Imposter Syndrome
⭐ ⭐⭐⭐

Lack of Purpose

Solutions trying to find problems, rather than good understanding of the problem

Feature Factory - project mindset hidden behind 'agile'

Lack of Rest

trendy technology ack choices
⭐ ⭐ ⭐ ⭐

Trying to solve social problem with technical solutions

Lack of trust
⭐⭐⭐⭐

Imposter Syndrome

Developers loving complexity

Lack of leadership

Lack of contact with real users - this bond is ultra important

Tracking utilization, leaving no room for learning and play

Tangible Metrics (i.e., is this really helping?)

Biz people do not understand the risk of software

Lack of trust