Software Requirements Engineering

Zhiyuan Wan

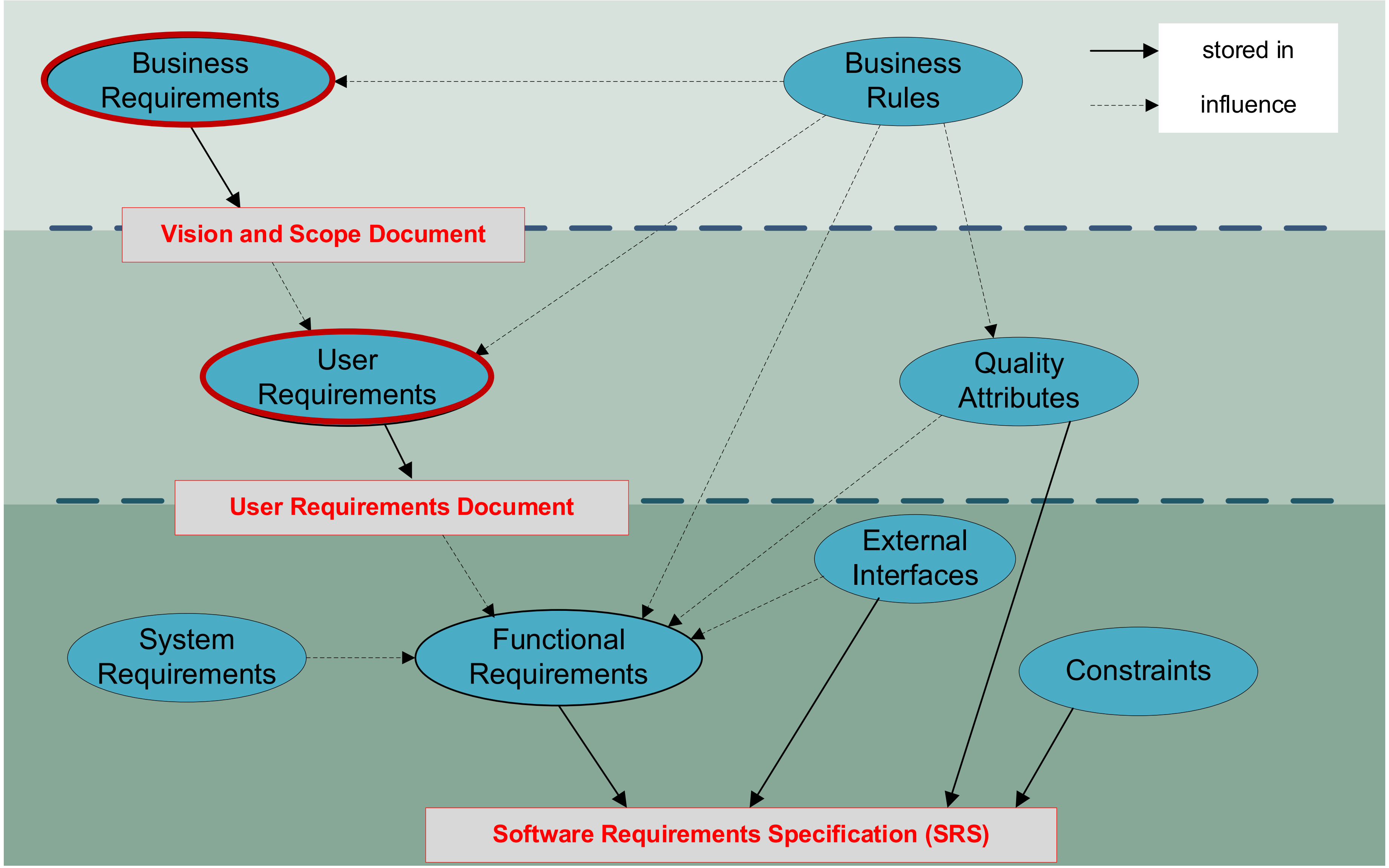
September 24, 2021

Agenda

- Recap from last lesson
- Build the knowledge
- In-class presentations
- Project

Previously on
software Requirement Engineering

Three levels of software requirements



Build the Knowledge

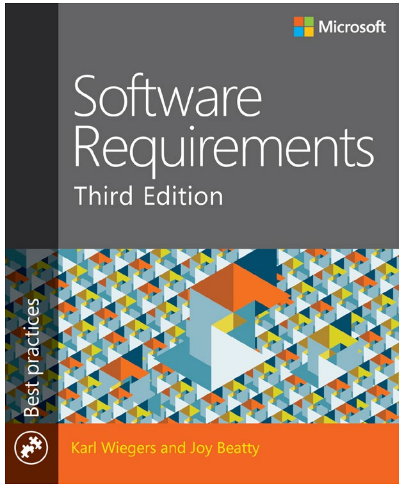
What we focus today

PART II: REQUIREMENTS DEVELOPMENT

CHAPTER 5 Establishing the business requirements

CHAPTER 6 Finding the voice of the user

CHAPTER 8 Understanding user requirements



Clearly Defined Business Requirements

Business Requirements

Business Objectives

- ✓ ***Come from business problems or market opportunities***
- ✓ ***Define success metrics for customer value***

Product Vision

- ✓***Concept of what the product might eventually become***
- ✓***Identifies business benefits of the system will provide***

Project Scope

- ✓ ***Boundary between what's in and out***
- ✓ ***Include limitations and exclusions***
- ✓ ***Facilitate making project commitments and priorities***
- ✓ ***Prerequisite to managing scope creep***

A Sample **Vision Statement**

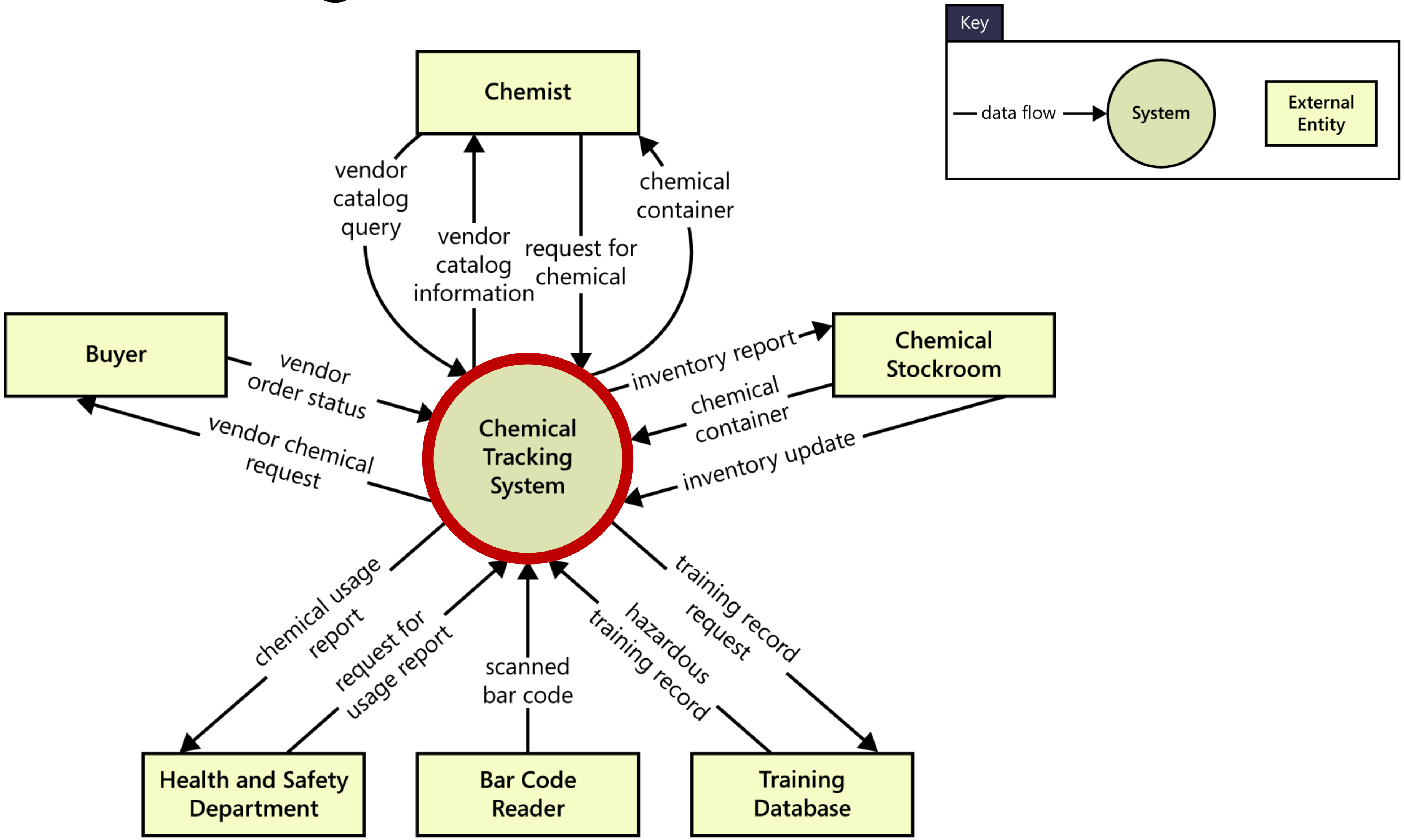
For chemists ***who*** need to request containers of chemicals, ***the*** Chemical Tracking System ***is*** an information system ***that*** will provide a single point of access to the chemical stockroom and to vendors. The system will store the location of every chemical container within the company, the quantity of material remaining in it, and the complete history of each container's locations and usage. This system will save the company 25 percent on chemical costs by letting the company fully exploit chemicals already available within the company, dispose of fewer partially used or expired containers, and use a single standard chemical purchasing process. ***Unlike*** the current manual ordering processes, ***our product*** will generate all reports required to comply with federal and state government regulations that require the reporting of chemical usage, storage, and disposal.

keyword template from Geoffrey Moore. Crossing the Chasm. HarperBusiness, 2002

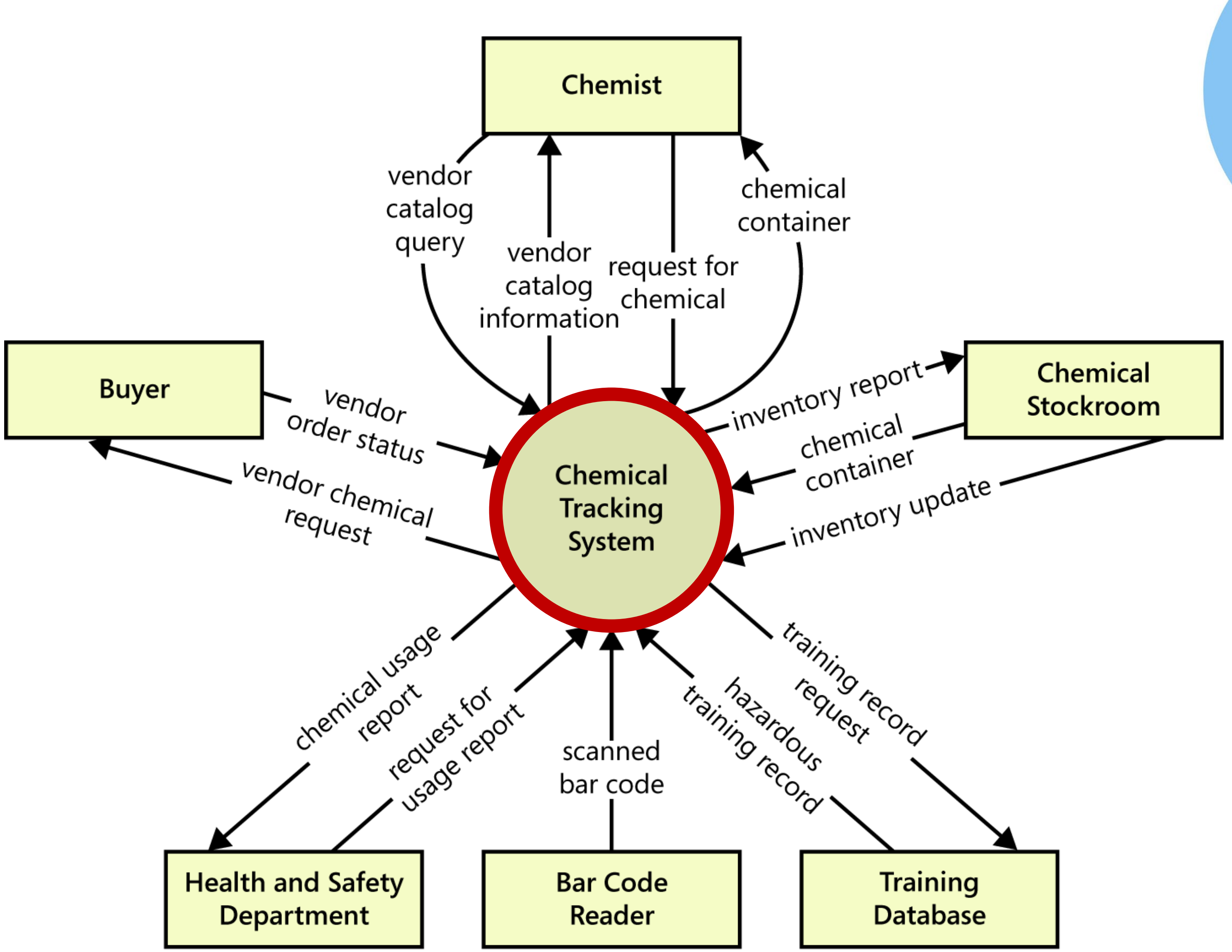
Four Techniques for **Scope** Definition

- ❑ Context Diagram
  - ✓ shows external entities
  - ✓ no internal system details
- ❑ Use Case Diagram
  - ✓ shows use case-actor connections
- ❑ Feature Roadmap
  - ✓ describe levels of feature enrichment
  - ✓ plan specific feature levels for each release
- ❑ System Events
  - ✓ triggers that stimulate some system response
  - ✓ could be business events, temporal events, input signals

The Context Diagram



The Context Diagram



Chemist

Buyer

Chemical Stockroom

Chemical Tracking System

Health and Safety Department

Bar Code Reader

Training Database

vendor catalog query

vendor catalog information

request for chemical

chemical container

vendor order status

vendor chemical request

inventory report

chemical container

inventory update

chemical usage report

request for usage report

scanned bar code

hazardous training record

training record request

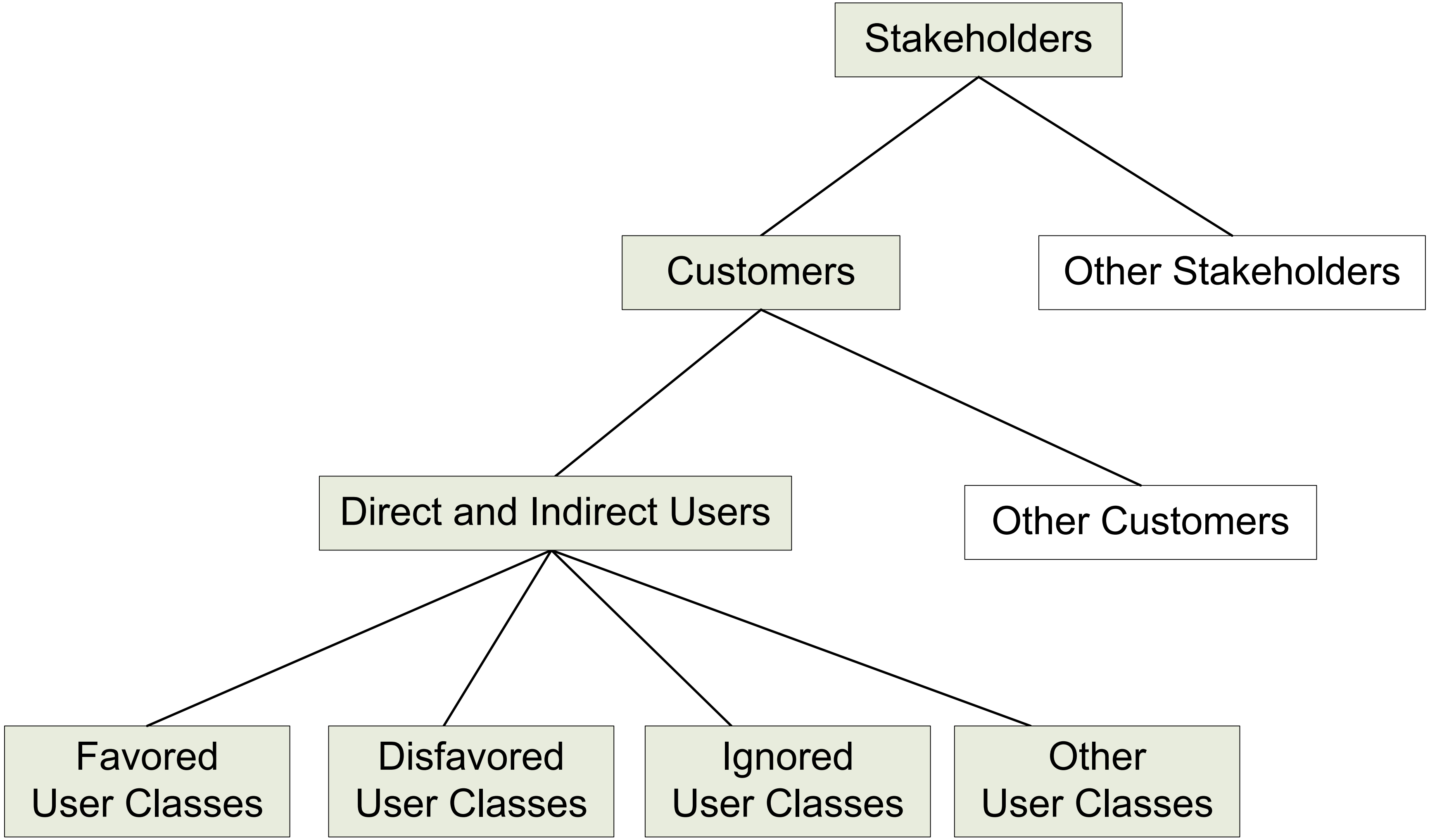
https://www.vecteezy.com/

“Why” Business Requirements

Using business objectives to determine completion

A Collaborative Partnership with Customers

From Stakeholders to User Classes



User Classes

- Distinct groups of users for a product
- Might differ in:
  - ✓ frequency of use of application
  - ✓ functions used
  - ✓ tasks to be accomplished
  - ✓ education and skill level
  - ✓ privilege or security level
- Identify user classes and their characteristics early
- Document user classes in the SRS
- Not all user classes are equally important to you
- What are your user classes? Which ones are favored?

Copyright © Randy Glasbergen. www.glasbergen.com

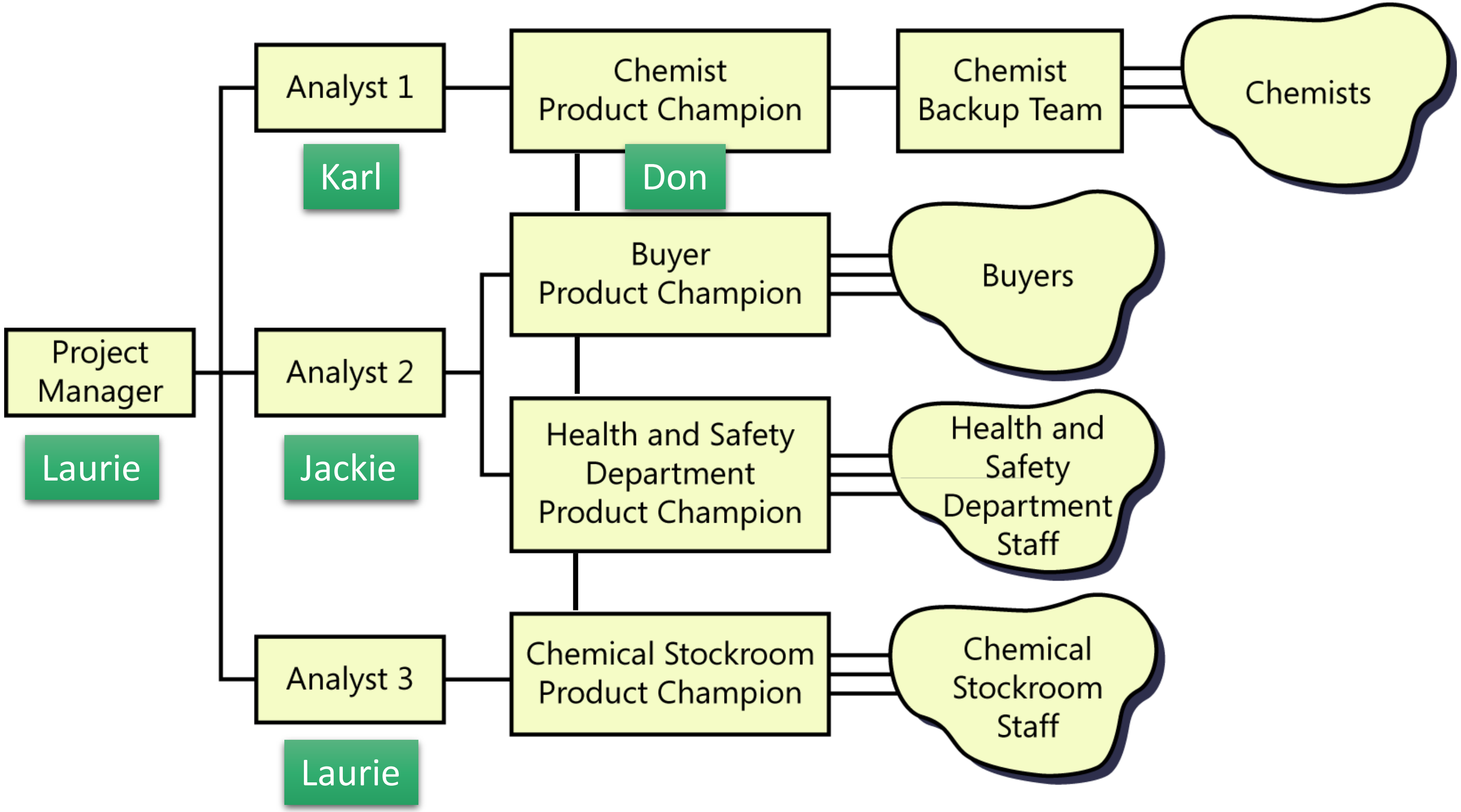
Gause and Lawrence, "User-Driven Design," Software Testing & Quality Engineering Jan/Feb 1999

The Product Champions

- ❑ Primary interface between development and customer communities
- ❑ Ideally, a real user; not a user manager, funding sponsor, or simulated user
- ❑ Reconciles incompatible user requirements
- ❑ Goal is to present developers with a unified set of requirements
- ❑ Must be empowered to make binding decisions
- ❑ Document product champion responsibilities

https://www.vectorstock.com/

Product Champions with Multiple User Classes



Understanding User Requirements

Use Cases Defined

Use Case: ***A description of a set of interaction sequences that a system performs to provide a result of observable or measurable value to one or more actors.***

❑ Use cases describe:

✓ user goals

✓ the user's view of the system

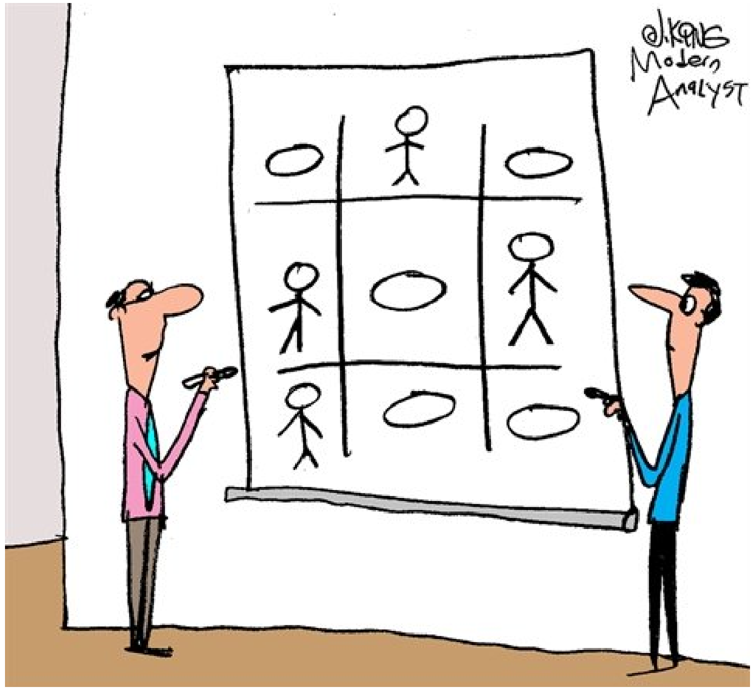
✓ a set of task-related activities

❑ Use cases do ***not*** describe:

✓ user interface designs

✓ technology solutions

✓ application architecture



https://www.modernanalyst.com/

Use Cases, Scenarios, and Stories

More Abstract

Prepare a mailing label.

Prepare a mailing label to send a package by second-day UPS.

Chris wants to send a 2.5-pound package by second-day UPS from Portland, OR, to Rochester, NY. She wants it insured for $75 and she wants a return receipt. The package is marked fragile.

Less Abstract

Postal Delight

https://www.gograph.com/

Use Cases and Functional Requirements

- Two schools of thought:
  - ✓ Use cases ***are*** the functional requirements.
  - ✓ Use cases ***reveal*** the functional requirements.

User View

CAN'T YOU DO ANYTHING RIGHT?
GLASBERGEN
Copyright © Randy Glasbergen. www.glasbergen.com

User Cases

Analyst

Developer View

Functional Requirements

Project

Project Evaluation

A group project of developing a web application in teams of 4-5 members.

There will be a series of milestones requiring group and individual reports (documents, GitHub repository, commit history, etc).

Your mark will be determined by

1) the quality of your deliverables of milestones
2) version control
3) how well your project is managed
4) and your mid-term and final presentations

Version Control

Steps

- Setup a private GitHub repository
- Invite all the team members in the GitHub repository
- Restrict document and code update access to authorized individuals
- Setup the conventions to commit changes to documentation and code (e.g., format of commit message, committer, version tag)
- Start tracking the progress since the first draft of the first output

Recommended tools

- Documentation – Markdown

Project Management

- Setup a “process” folder in GitHub repository
- Make plan for the project
- Reach agreement on the communication strategy, e.g., weekly meeting time and format
- Prepare template for meeting minutes

Recommended tools

Notion

Milestones in the Project

ID	Milestone	Percentage	Deliverable (ZIP, PDF)	Due
1	Team Workflow	5%	**1. Document: (PDF)** <= 5 pages **2. Process files: (ZIP)** Snapshots of online meetings, meeting minutes, commit logs of the GitHub repo **3. Presentation: (PDF)** <= 15 minutes Transfer slides to PDF	24:00 Oct 6, 2021
2	Vision and Scope	10%		24:00 Oct 13, 2021
3	Software Requirements Specification and Mid-Term Presentation	20%		24:00 Nov 3, 2021
4	Design and Coding	35%		24:00 Dec 22, 2021
5	Final Report and Presentation	30%		24:00 Dec 29, 2021

Email to: zju_sre_21@163.com before due date;
Naming: PROJECT-MILESTONE[n]-[team_id]-[team_name]

Milestone 1 Team Workflow

The team workflow is a document, including:

- **Team name:** Figures out one unique name.
- **Roles:** Outlines team roles and responsibilities.
- **Strategies:** Gives us information about strategy for project management, e.g., team meeting times (weekly meeting, who will arrange the meeting and record meeting minutes)
- **Plan:** Makes timeline for each milestone, making sure that buffers are included in your plan and at least two iterations are scheduled for each milestone.

The main purpose of this document is to give you some rules for team process, management, tracking, and goal setting.

In-Class Presentation

Homework 1: Task 3 | Questions and Answers

HW 1 – Submission (updated)

Task	Content Included	Format	Due
Task 1: Find out 3 papers	For each paper: 1. Title, author(s), venue 2. Summary 3. Explain why it is interesting	**Zip file, in which there is** a) PDF (<=1 page), compiled from b) tex file c) bib file (BibTex of the three papers)	24:00 Oct 7, 2021
Task 2: Paper Reading	Explain your understanding of the paper	**PDF slides** (<=10 pages)	24:00 Oct 7, 2021 Notify me before 24:00 Oct 5, 2021 if you would like to give a presentation ☺
Task 3: Try the replication package of FRMiner	1. The process 2. The findings	**PDF slides** (<=10 pages)	24:00 Sep 23, 2021 Notify me before 24:00 Sep 21, if you would like to give a presentation ☺

Email to: zju_sre_21@163.com before due date;
Naming: HW[n]-TASK[m]-[student_number]-[full_name]

Latex Template

- Use the official “ACM Primary Article Template”, as can be obtained from the ACM Proceedings Template page (https://www.acm.org/publications/proceedings-template)
- LaTeX users should use the `sigconf` option.
- To that end, the following LaTeX code can be placed at the start of the LaTeX document:

```
\documentclass[sigconf]{acmart}
```

Reference:

https://conf.researchr.org/track/icse-2022/icse-2022-papers

More about the project

For business requirements

Software Development Analytics

Understand the software development projects that matter to you

https://bitergia.com

Business Requirements

For Software Development Teams

Evaluate project's success, guide investment decisisions and advocate the work that software development teams are doing

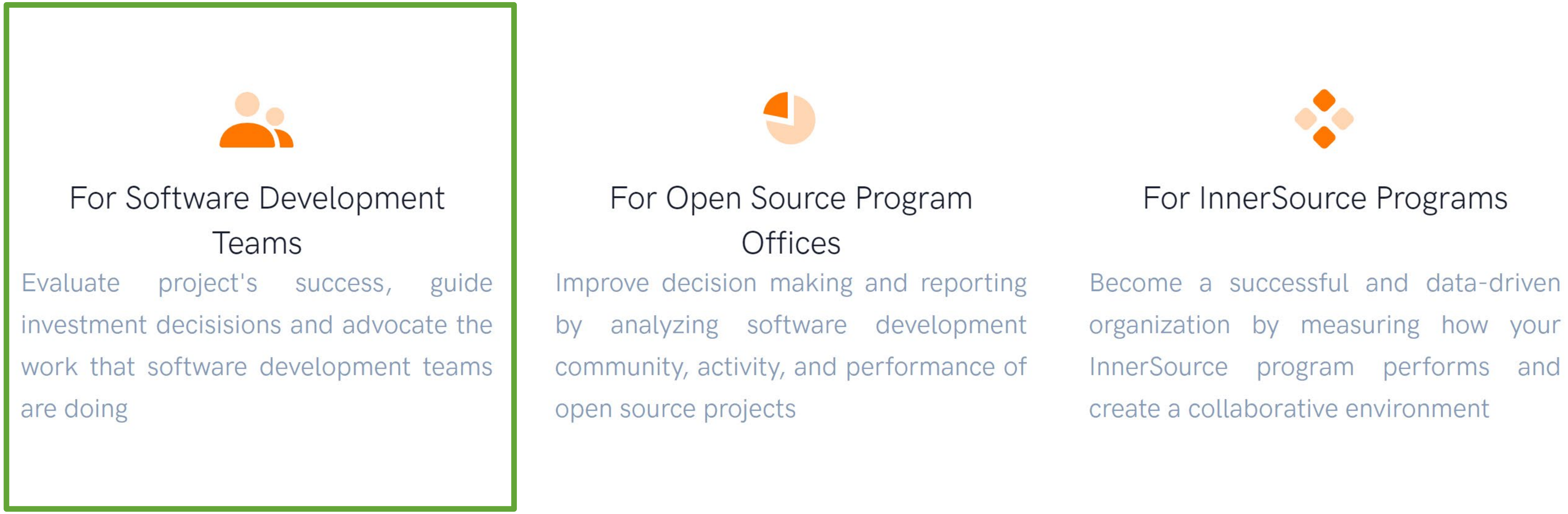
For Open Source Program Offices

Improve decision making and reporting by analyzing software development community, activity, and performance of open source projects

For InnerSource Programs

Become a successful and data-driven organization by measuring how your InnerSource program performs and create a collaborative environment

Business Requirements



For Software Development Teams

Evaluate project's success, guide investment decisisions and advocate the work that software development teams are doing

For Open Source Program Offices

Improve decision making and reporting by analyzing software development community, activity, and performance of open source projects

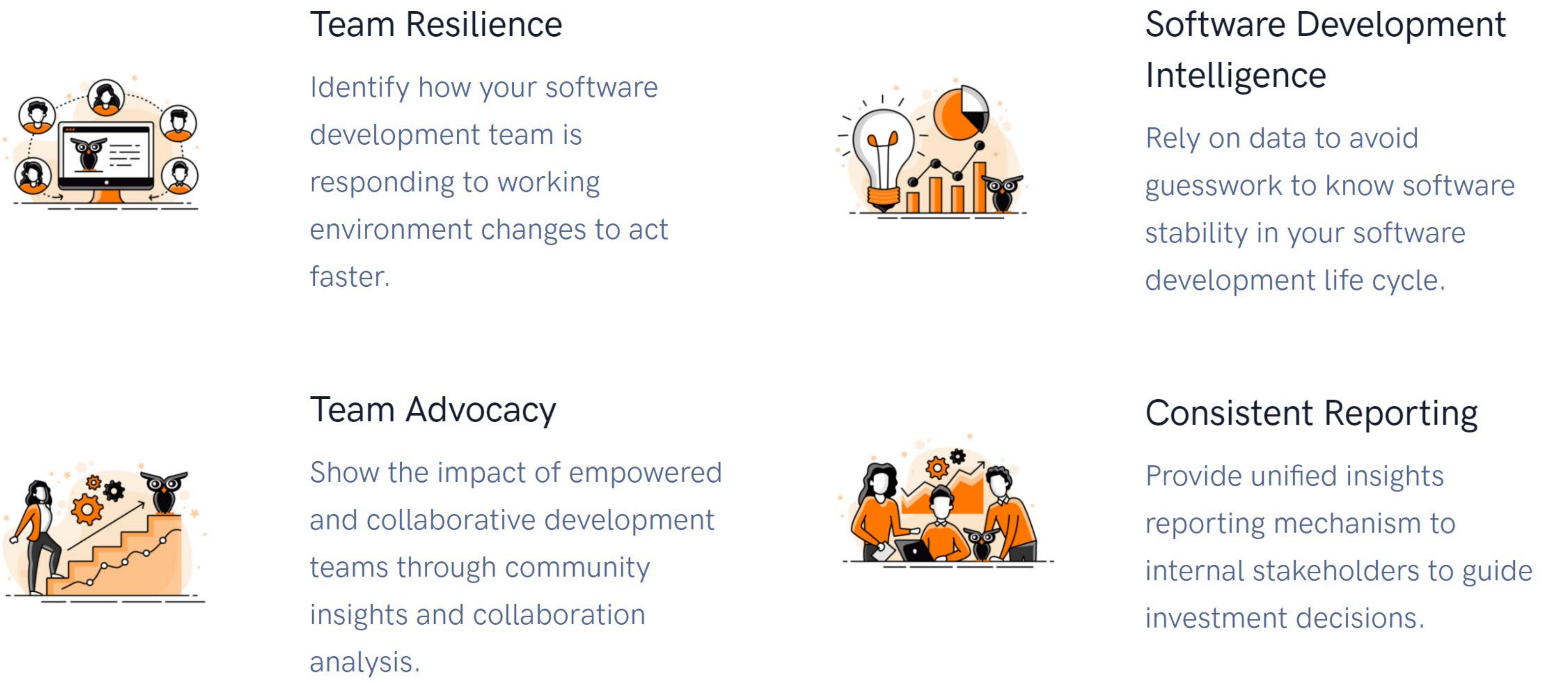
For InnerSource Programs

Become a successful and data-driven organization by measuring how your InnerSource program performs and create a collaborative environment

For Software Development Teams

Understand the business value of software development teams

Managers can evaluate project's success, guide investment decisions and advocate the work that software development teams are doing.



Team Resilience

Identify how your software development team is responding to working environment changes to act faster.

Software Development Intelligence

Rely on data to avoid guesswork to know software stability in your software development life cycle.

Team Advocacy

Show the impact of empowered and collaborative development teams through community insights and collaboration analysis.

Consistent Reporting

Provide unified insights reporting mechanism to internal stakeholders to guide investment decisions.

Business Requirements

For Software Development Teams

Evaluate project's success, guide investment decisisions and advocate the work that software development teams are doing

For Open Source Program Offices

Improve decision making and reporting by analyzing software development community, activity, and performance of open source projects

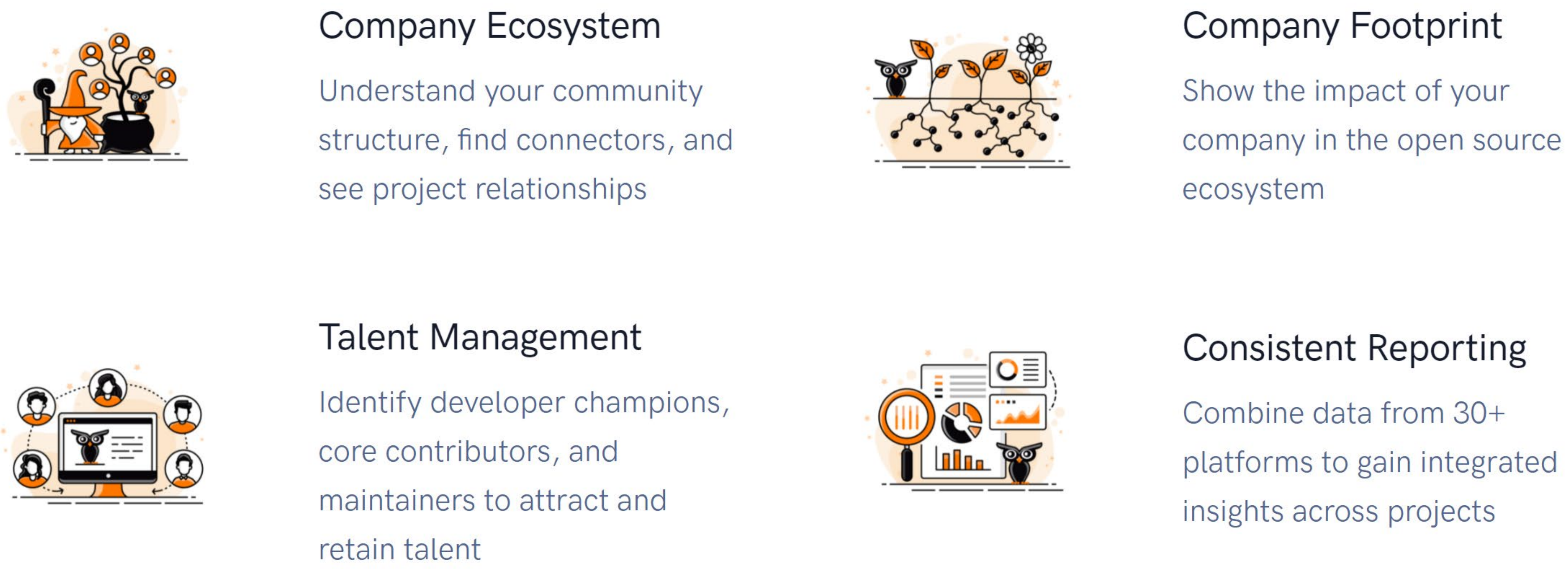
For InnerSource Programs

Become a successful and data-driven organization by measuring how your InnerSource program performs and create a collaborative environment

For Open Source Program Offices

Maximize ROI from Open Source Programs

Open Source Program Offices (OSPO) use the tool as a single source of truth about their ecosystem of open source projects.



Company Ecosystem

Understand your community structure, find connectors, and see project relationships

Company Footprint

Show the impact of your company in the open source ecosystem

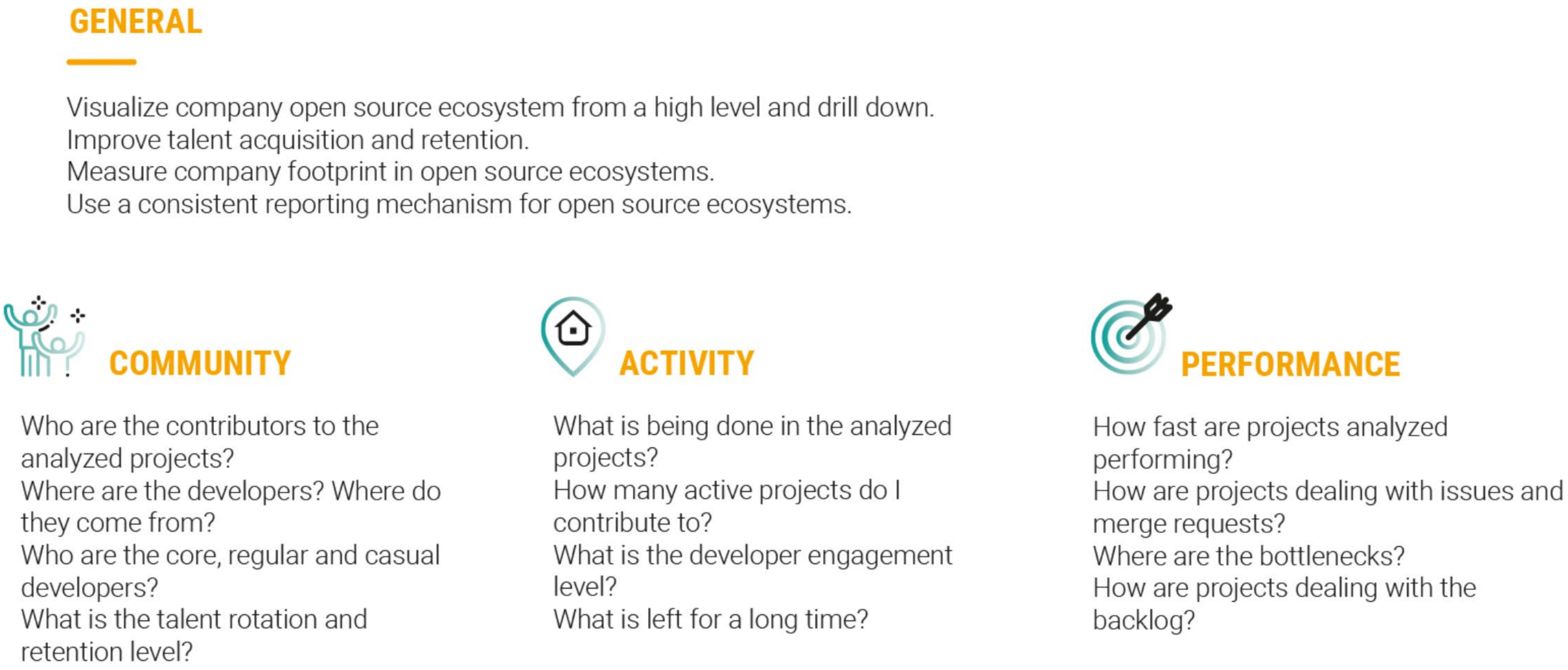
Talent Management

Identify developer champions, core contributors, and maintainers to attract and retain talent

Consistent Reporting

Combine data from 30+ platforms to gain integrated insights across projects

For Open Source Program Offices



GENERAL

Visualize company open source ecosystem from a high level and drill down.
Improve talent acquisition and retention.
Measure company footprint in open source ecosystems.
Use a consistent reporting mechanism for open source ecosystems.

COMMUNITY

Who are the contributors to the analyzed projects?
Where are the developers? Where do they come from?
Who are the core, regular and casual developers?
What is the talent rotation and retention level?

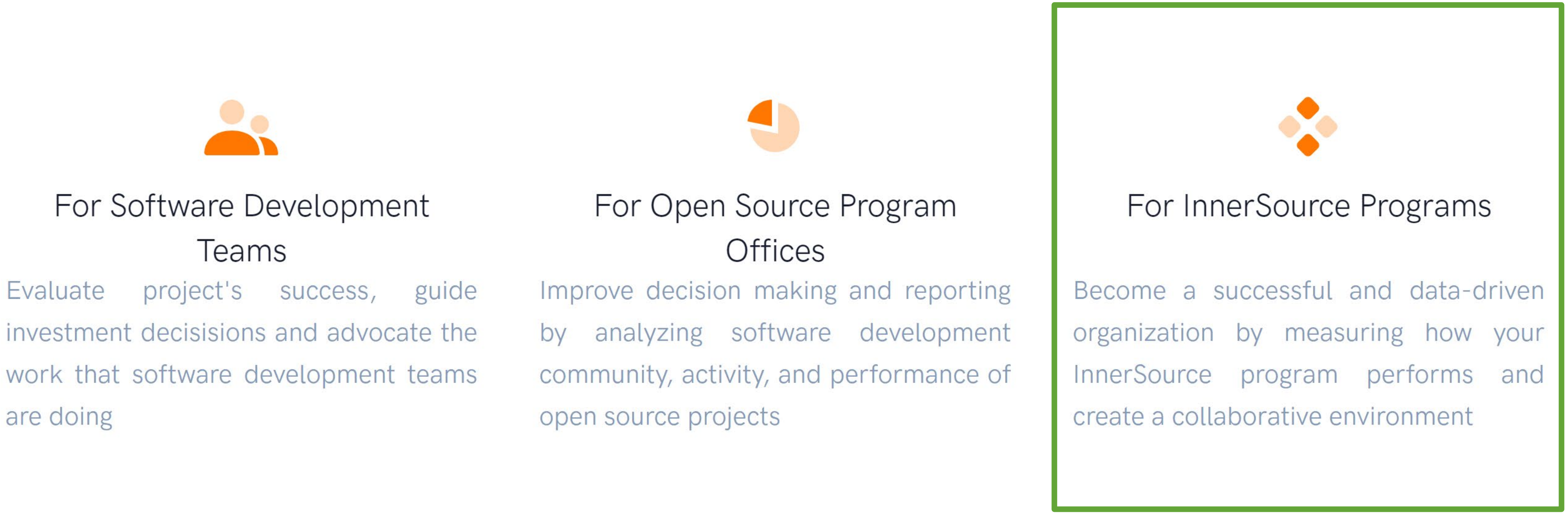
ACTIVITY

What is being done in the analyzed projects?
How many active projects do I contribute to?
What is the developer engagement level?
What is left for a long time?

PERFORMANCE

How fast are projects analyzed performing?
How are projects dealing with issues and merge requests?
Where are the bottlenecks?
How are projects dealing with the backlog?

Business Requirements



For Software Development Teams

Evaluate project's success, guide investment decisisions and advocate the work that software development teams are doing

For Open Source Program Offices

Improve decision making and reporting by analyzing software development community, activity, and performance of open source projects

For InnerSource Programs

Become a successful and data-driven organization by measuring how your InnerSource program performs and create a collaborative environment

For InnerSource Programs

Foster Innovation and Collaboration with InnerSource

The tool tracks community processes inside organizations to help digital transformation and breaking team silos.

Understand Engagement

Identify developer activity within your InnerSource projects to see trends and find most active or abandoned projects

Evangelization Support

Leverage best practices and lessons learned, backed by data, to advance your InnerSource initiative

InnerSource Maturity

Understand how mature the InnerSource practices, processes, and culture are while identifying areas worth improving

Collaboration Dynamics

Demonstrate how silos are broken down by visualizing collaboration across projects, teams, and departments

For InnerSource Programs

GENERAL

Foster collaboration, break silos.
How people are contributing in the InnerSource ecosystem and how this is helping to connect people.
Foster innovation through collaboration.
Increase the efficiency in the process.
Reusability of the software.

COLLABORATION

Who are the contributors in our projects?
Where are the developers from?
Who are the core, regular, and casual developers?
How effective is our business units cross-collaboration?

GROWTH & ENGAGEMENT

What projects do our business units contribute to?
What is our developer engagement and talent retention?
How many active projects does my business units contribute to?
What is the level of engagement of our different business units?

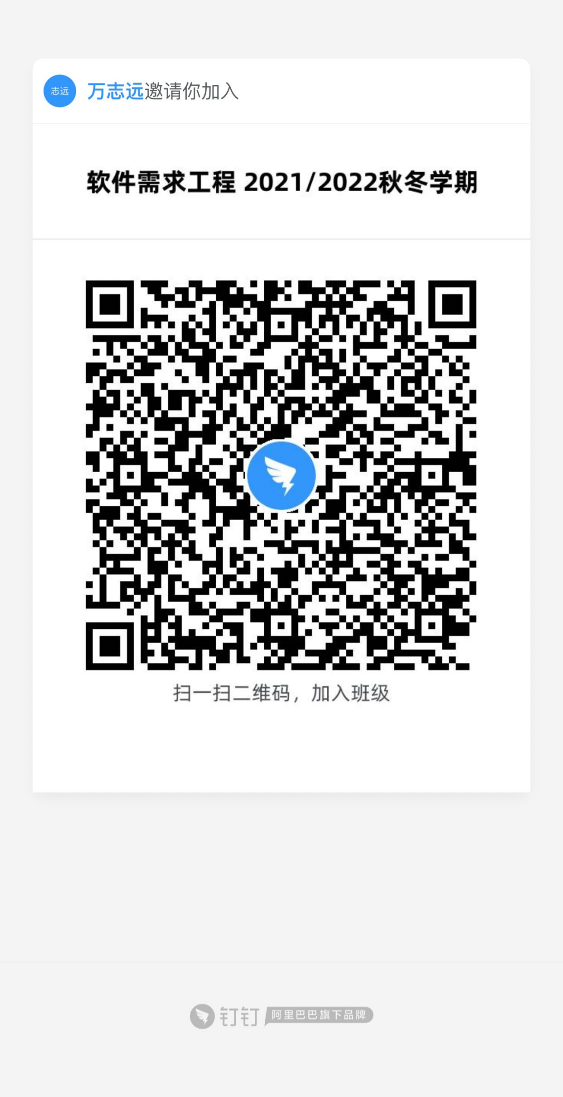
EFFICIENCY AND PROCESS COMPLIANCE

How performant are projects?
How are projects dealing with issues and merge requests?
Where are bottlenecks?
How are projects dealing with their backlogs?
Are we improving knowledge management, specifically reuse and optimization?
Are we speeding up onboarding?

All about you

还未（提前）组队的同学

我是谁？
我有啥技能？ 这些技能将为我的团队提供怎样的帮助？
我打算每周贡献多少时间？
我愿不愿意当Team Leader？ 希望在团队中担任怎样的角色？
我希望和什么样的同学合作？



万志远邀请你加入

软件需求工程 2021/2022秋冬学期

扫一扫二维码，加入班级