AI Team Insights Prompt Library

Surfacing Patterns That Take Humans Weeks to Discover

A companion resource from Agile Meets AI, Chapter 7

AI can analyze team patterns from commit history to delivery data, surfacing insights that inform how high-performing teams work. These prompts help you discover which parts of the codebase cause problems, where your team spends time, what patterns correlate with success, and where skill gaps create vulnerability.

Codebase Risk Analysis

Identify which parts of your codebase cause the most bugs and where complexity creates risk.

Data needed: Git log export + bug/defect data with affected files (see data gathering instructions below)

```
Analyze this data to identify high-risk areas in our codebase.

[PASTE GIT COMMIT DATA HERE - Include: file paths, commit frequency, authors]

[PASTE DEFECT DATA HERE - Include: defect ID, severity, files affected]

Please provide:
1. HOT SPOTS: Which files or directories have the most bug-related commits?
2. COMPLEXITY RISK: Which areas change frequently AND have defects?
3. OWNERSHIP PATTERNS: Are problem areas owned by a few or many contributors?
4. CORRELATION ANALYSIS: Do bugs cluster in specific types of files (tests, config, specific modules)?
5. RISK RANKING: Rank the top 10 highest-risk areas by combined frequency + severity.
6. RECOMMENDATIONS: Where should we focus on refactoring, testing, or documentation efforts?

Identify any files that are both frequently changed AND frequently associated with defects—these are priority targets.
```

💡 **Tip:** *Export git log with: git log --pretty=format:'%h,%an,%ad,%s' --name-only --since='6 months ago' > commits.csv*

Time & Effort Pattern Analysis

Understand where your team spends time versus where they think they spend time.

Data needed: Work item data with time tracking or cycle time by component/area

```
Analyze where our team spends the most effort.

[PASTE WORK ITEM DATA HERE - Include: item ID, type, component/area, story points or time spent, status]

[OPTIONAL: PASTE COMMIT DATA BY AREA - file paths and commit counts]

Please provide:
1. EFFORT DISTRIBUTION: What percentage of effort goes to each component/area?
2. WORK TYPE BREAKDOWN: How much time on features vs bugs vs tech debt vs maintenance?
3. HIDDEN TIME SINKS: Which areas consume more effort than their business value suggests?
4. PATTERN ANALYSIS: Are there areas where small items take disproportionately long?
5. CHURN INDICATORS: Which components require repeated rework?
6. EFFICIENCY OPPORTUNITIES: Where could focused improvement free up the most capacity?

Compare effort allocation against stated priorities—are we spending time where it matters most?
```

💡 **Tip:** *If you don't have time tracking, use story points by component as a proxy, or commit counts by directory.*

Delivery Success Patterns

Discover what conditions correlate with successful deliveries versus problematic ones.

Data needed: Sprint/release data with outcomes, plus contextual factors

```
Analyze our delivery history to identify success patterns.

[PASTE DELIVERY DATA HERE - Include for each sprint/release:
- Sprint/release ID and dates
- Committed vs completed items
- Escaped defects count
- Team size and any changes
- Notable events (holidays, incidents, team changes)
- Subjective success rating if available (1-5)]

Please provide:
1. SUCCESS FACTORS: What conditions are present in our best deliveries?
2. FAILURE PATTERNS: What conditions correlate with problematic releases?
3. TEAM STABILITY IMPACT: How do team changes affect delivery success?
4. CAPACITY PATTERNS: Is there an optimal commitment level for this team?
5. EXTERNAL FACTORS: Do holidays, dependencies, or other factors predict problems?
6. PREDICTIVE INDICATORS: What early signs suggest a delivery is at risk?

Identify 3-5 actionable insights we can use to increase delivery success rate.
```

💡 **Tip:** *Include both quantitative data (velocity, defects) and qualitative notes (what made this sprint feel good or bad).*

Team Skill & Vulnerability Analysis

Map team skills to identify single points of failure and T-skill development opportunities.

Data needed: Team roster with skills/areas of expertise, plus commit or work history by person and area

```
Analyze our team's skill distribution and identify vulnerabilities.

[PASTE TEAM SKILLS DATA HERE - For each team member:
- Name/ID
- Primary skills/specialties
- Secondary skills (areas they can contribute)
- Areas they're learning or interested in]

[PASTE WORK DISTRIBUTION DATA HERE - commits or work items by person and component/area]

Please provide:
1. SKILL COVERAGE MAP: Which areas are well-covered vs single-person dependent?
2. BUS FACTOR ANALYSIS: Where would losing one person create critical gaps?
3. T-SKILL OPPORTUNITIES: Who is close to being able to contribute to adjacent areas?
4. NATURAL PAIRINGS: Which team members could mentor each other to build redundancy?
5. LEARNING PRIORITIES: Which skill gaps are most critical to address?
6. DEVELOPMENT PATHS: Suggest specific growth paths for building team resilience.

Identify our top 3 vulnerability areas and recommend specific actions to address each.
```

💡 **Tip:** *If you don't have formal skills data, use git blame or work item assignments to infer who knows what.*

Quick Data Gathering Commands

Git commit history (last 6 months):

```
git log --pretty=format:'%h,%an,%ad,%s' --name-only --since='6 months ago'
```

File change frequency:

```
git log --name-only --pretty=format: | sort | uniq -c | sort -rn | head -50
```

Contributors by file/directory:

```
git shortlog -sn --all -- path/to/directory/
```

For Jira/Azure DevOps: Use built-in export features to CSV, or query APIs for work item history.

> **Remember:** AI surfaces patterns—humans decide what they mean and what to do about them. Use these insights to spark team conversations, not to assign blame or make unilateral decisions.

From *Agile Meets AI: A Pragmatic Guide for Modern Teams* by Sheila Eckert
Download more resources at thesheilaverse.com/book-companion