

研发效能的数字可视化、洞见与改善

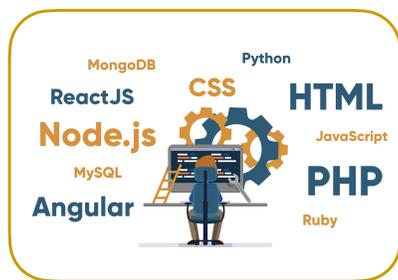
张思楚

ThoughtWorks Technical Principal

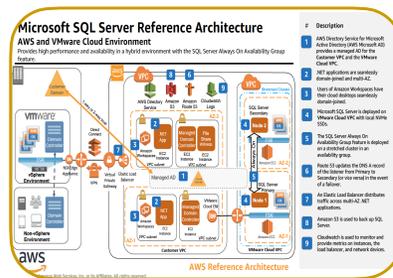


张思楚
Technical Principal

海外大项目的离岸交付中心技术负责人，全栈工程师、畅销Web产品SpreadWeb架构师，3项Web专利技术发明人。目前主要关注于软件交付效能，有效实施持续集成和持续交付，落地合适的软件效能度量，帮助大型团队通过有效的效能度量和管理，保证团队稳定高效的项目交付。



Full Stack Developer



Architect



Technical Principal



100
摘要

- ★ 客户希望与我们建立离岸交付中心的合作模式，开发并交付业务功能。
- ★ 如何可视化交付中心的交付效能，可视化各个团队的工作状态。
- ★ 如何识别阻塞交付效能的各种因素，从而针对性的治理，形成高效能的交付团队。
- ★ 本案例将系统化的介绍整个过程中的经验和教训，帮助您提升团队的交付效能。

研发效能的数字可视化、洞见与改善

软件研发过程的数字化、可视化、让软件从业者终于有机会可以度量研发过程中各个阶段的效能。度量这些数字是让人兴奋的，但是如何让它真正帮助到实践者？

- ★ 真正的一线高手，似乎并不真正在意度量的结果，即便把数据做了分析报告，但很少有人看，或者看不出问题。
- ★ 在报告上研发效能已经达到了指标，但是实际的业务结果却没有明显改善？这该如何是好？
- ★ 为了解决这些问题，我们需要获取并度量哪些数据？
- ★ 如何制定合适的指标，用指标帮助解决实际的问题，识别并治理技术风险和交付风险？
- ★ 如何最终通过度量解决实际的研发问题，增强研发效能，带来**业务成果**的正向反馈？

案例背景

一位海外客户希望与我们建立离岸交付中心（百人左右的离岸交付中）的合作模式，在交付中心开发并交付业务功能，在合作中客户主要关心交付的效能和质量。

如何可视化交付中心的研发效能，采用什么方式度量，可视化各个团队的工作状态，识别阻塞研发效能的各种因素，从而针对性的给出治理措施，保证质量，形成高效能的交付团队？这成为了让客户满意，治理离岸交付中心的工作重点。

参考业界的经验，团队不断的实践，总结出了两个主要维度的7个指标，这次案例将分享这个7个指标的实践和思考。

为什么研发效能需要数字化？

★ 德鲁克、戴明：If You Can't Measure It, You Can't Improve It.

★ 开尔文（英国物理学家）：当你能够量化你谈论的事物，并且能用数字描述它时，你对它就确实有了深入了解。

但如果你不能用数字描述，那么你的头脑根本就没有跃升到科学思考的状态。

★ “经验”未必可靠，你能看到“真相、本质”吗？



抓手 抓手 抓手

If You Can't Measure It, You Can't Improve It 不等价于 If You Can Measure It, You Can Improve It

Measuring something doesn't make it manageable. It just makes it measurable.



抓手 可以进行改进和管理的抓手

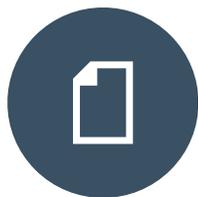
TOP1

TOP100 研发效能怎么数字化?



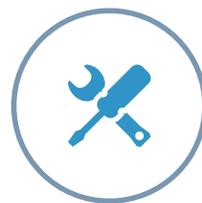
数据采集

- ★ 研发过程数据
- ★ 研发成本数据
- ★ 阻塞跟踪数据
- ★ 会议时长数据
- ★ 数据合规数据



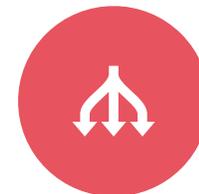
数据分析

- ★ 指标有效性?
- ★ 策略合理性?
- ★ 回馈组织OKR/KPI



建立模型

- ★ 效率模型
- ★ 质量模型
- ★ 业务模型
- ★ 成本模型



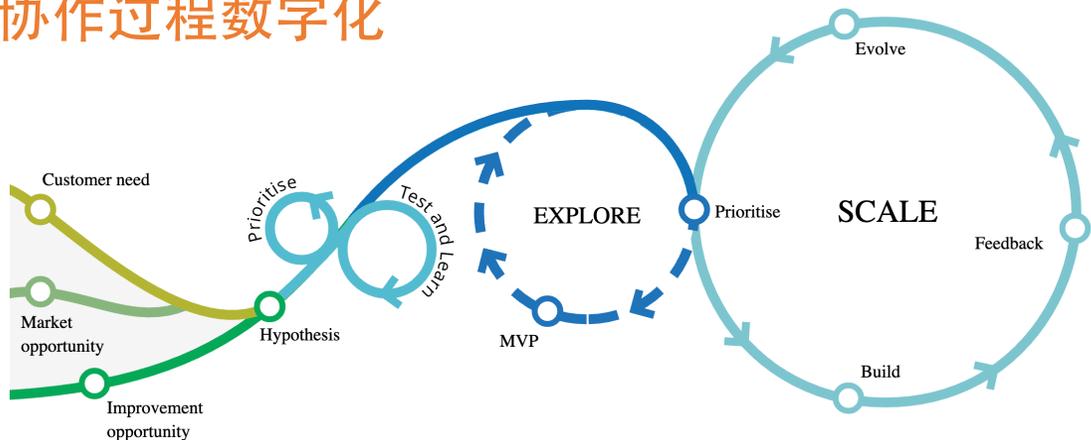
辅助决策

- ★ 效能同比/环比
- ★ 效能趋势/预测
- ★ 效能辅助决策
- ★ 效能回馈业务

研发效能数字化的常规过程

研发效能数字化的土壤已基本就绪

协作过程数字化



开发过程数字化

```

* d82360d (HEAD, origin/master, origin/HEAD, master) Added current class in both html- and css-file
* 5f11b48 (origin/pu/dropdown, pu/dropdown) Merge branch 'pu/search-field'

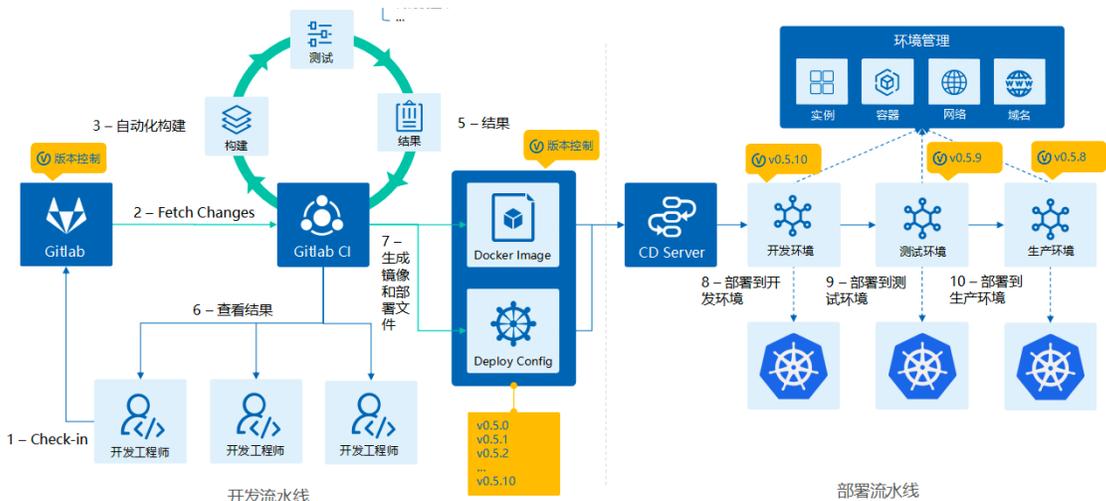
* 0bd1fe9 (origin/pu/search-field, pu/search-field) Implemented the search-field's background-image
* 326dea6 Changed height of search field to 20px
* 170be1d Coloured background of search field
* 5e16dab Merge branch 'jquery' into pu/search-field

* ae93982 (origin/pu/jquery, origin/jquery) Implemented the 'focus' class
* cae0c85 Changed the file names of all jquery scripts
* 5414a35 Using jquery to implement placeholder-attribute
* 5c5cde8 Merge branch 'jquery' into pu/search-field

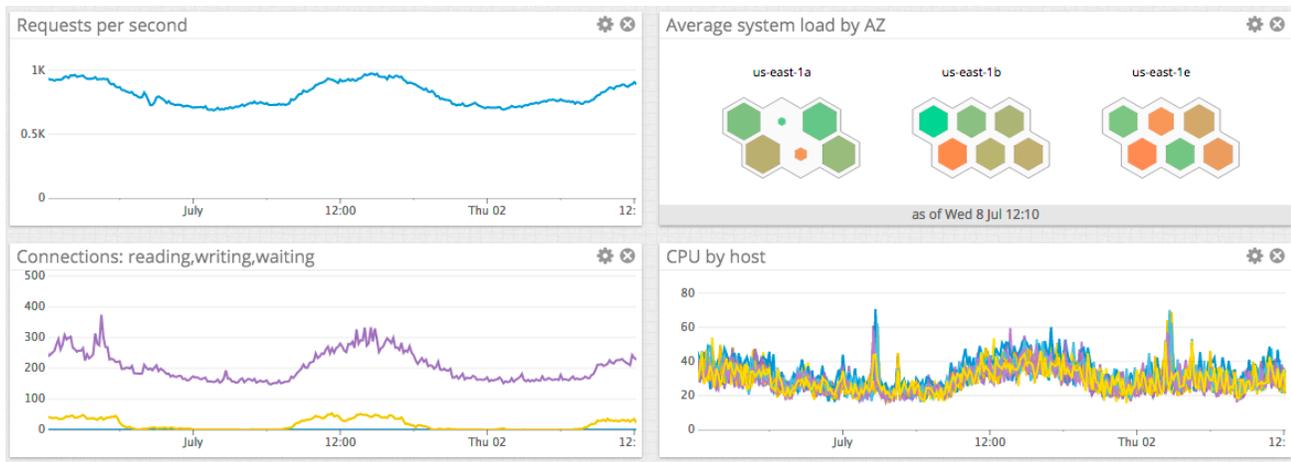
* 339daeb (origin/pu/vector, pu/vector) Changed search-field's length to 130px
* 7e5c6bd Added SVG search field in site/img/svg/search.svg
* fb90b1b Altered the root item's css design
* 346a626 Fixed a bug in the main navigation

* 4f772ee Changed copyleft-image to rotated copyright symbol
    
```

工程过程与制品数字化



运行状态数字化



在这个案例里我们是怎么做到的？

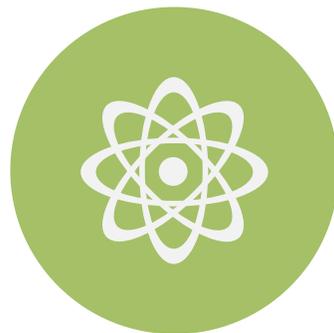


协作维度

速率/吞吐量

功能交付时长/阻塞时长

交付功能百分比



工程维度

变更前置时长

部署频率

上线失败率

平均恢复时长



协作维度

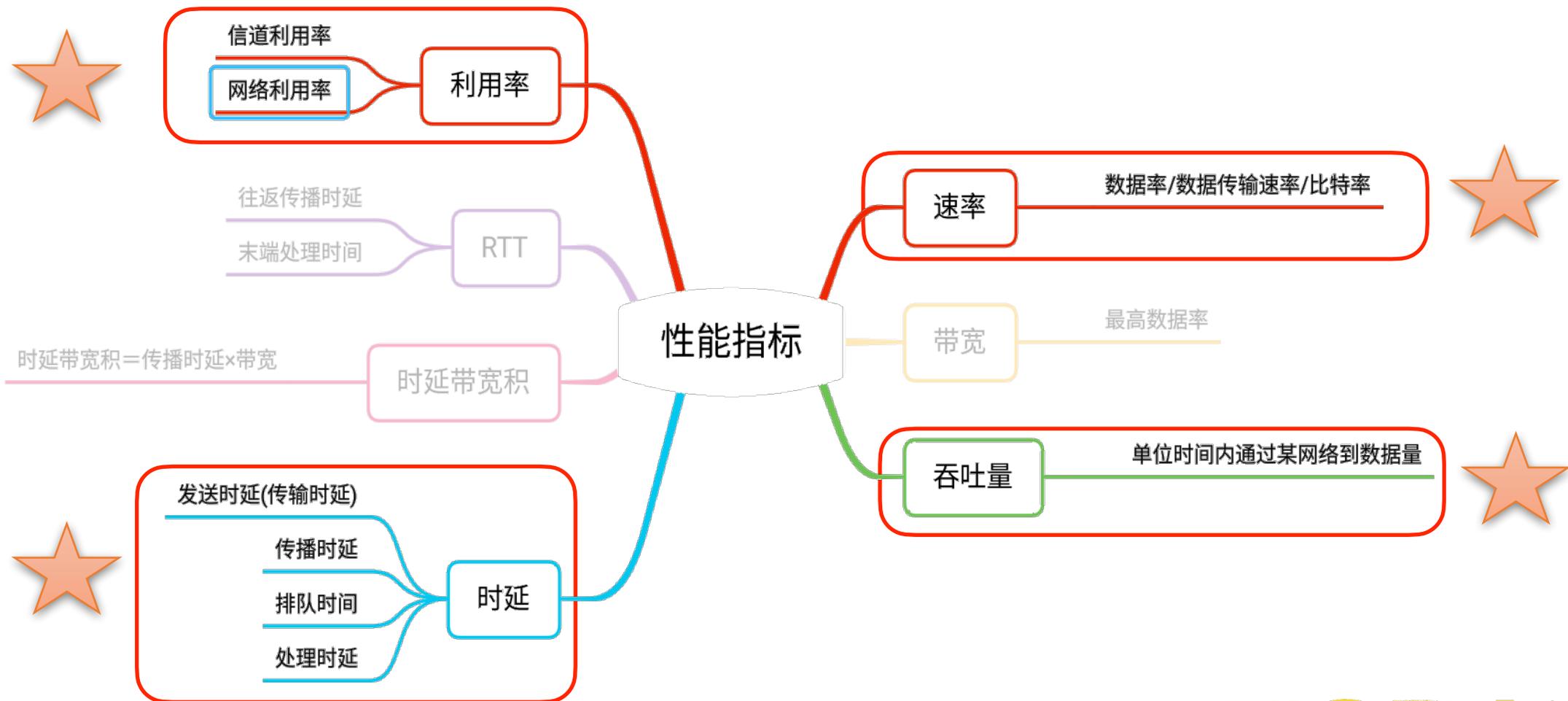
速率/吞吐量

功能交付时长/阻塞时长

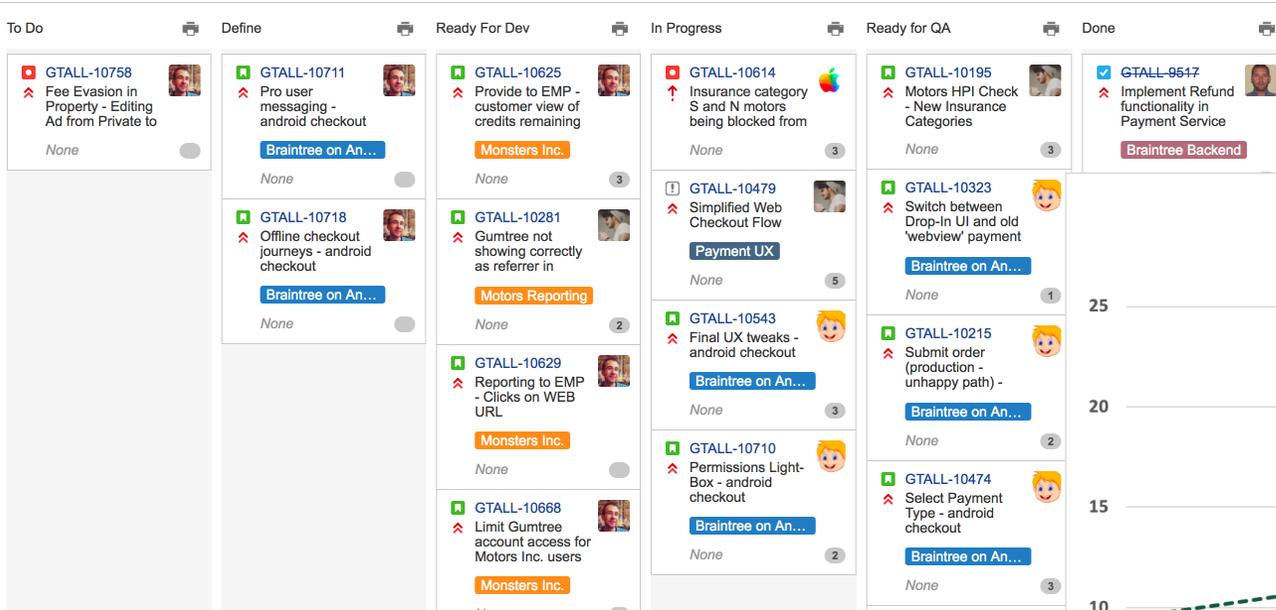
交付功能百分比

实践-协作维度-速率/吞吐量

计算机网络的性能指标是否能给我们一些灵感?

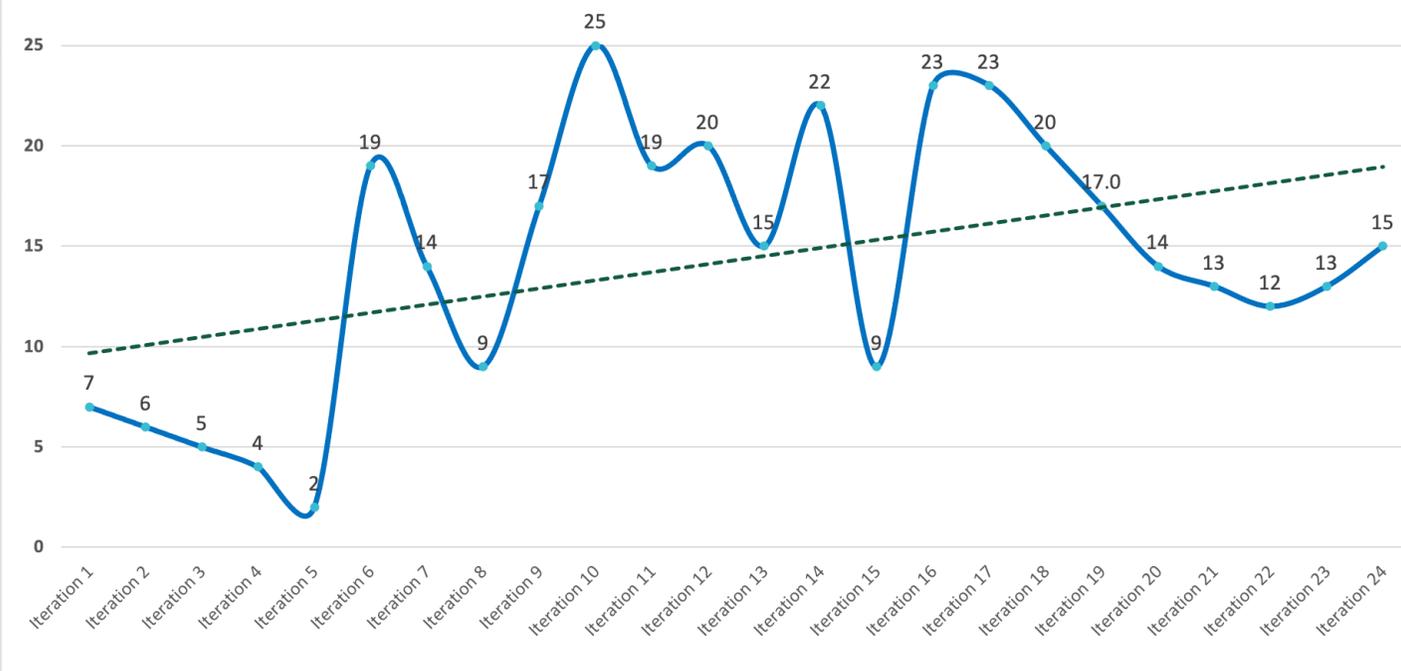


实践-协作维度-速率/吞吐量



速率/吞吐量 - 每个迭代完成的点数/卡数

Throughput - Completed cards by sprint



Velocity/throughput dropped quickly because TL took a week off, 5 points in testing that will contribute to next sprint's velocity.

★ 你是否连续跟踪过团队多个迭代的速率/吞吐量?

★ 不同团队的迭代速率/吞吐量是否可以比较?

★ 他们有时高或者有时低的原因你分析过吗?

★ 什么才是合适的趋势?

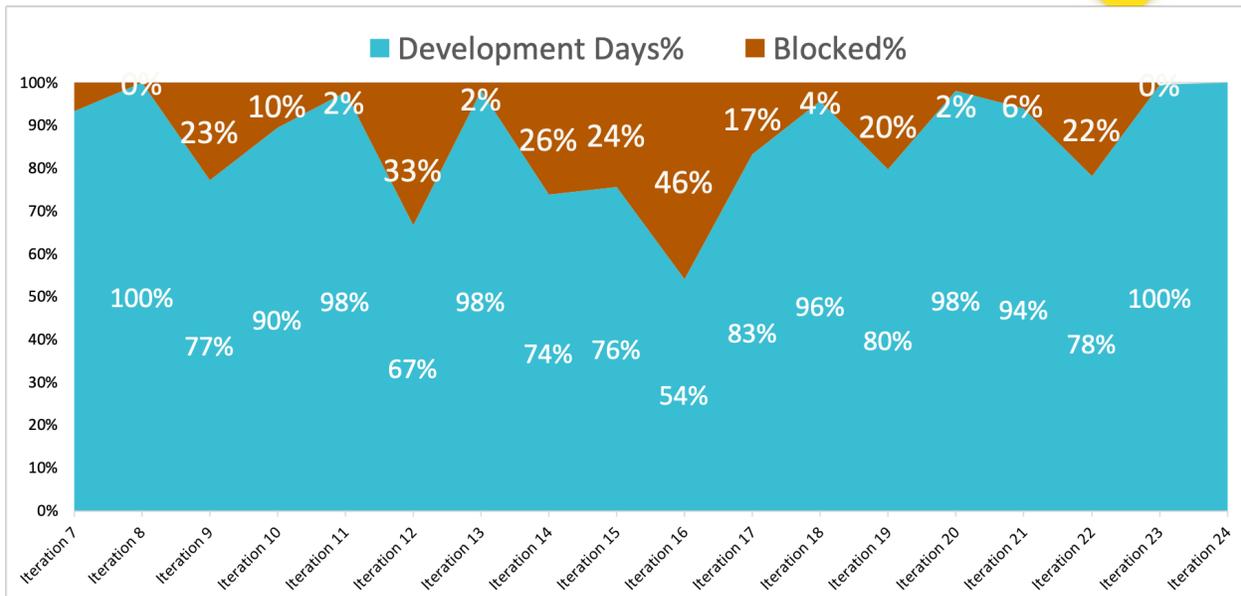
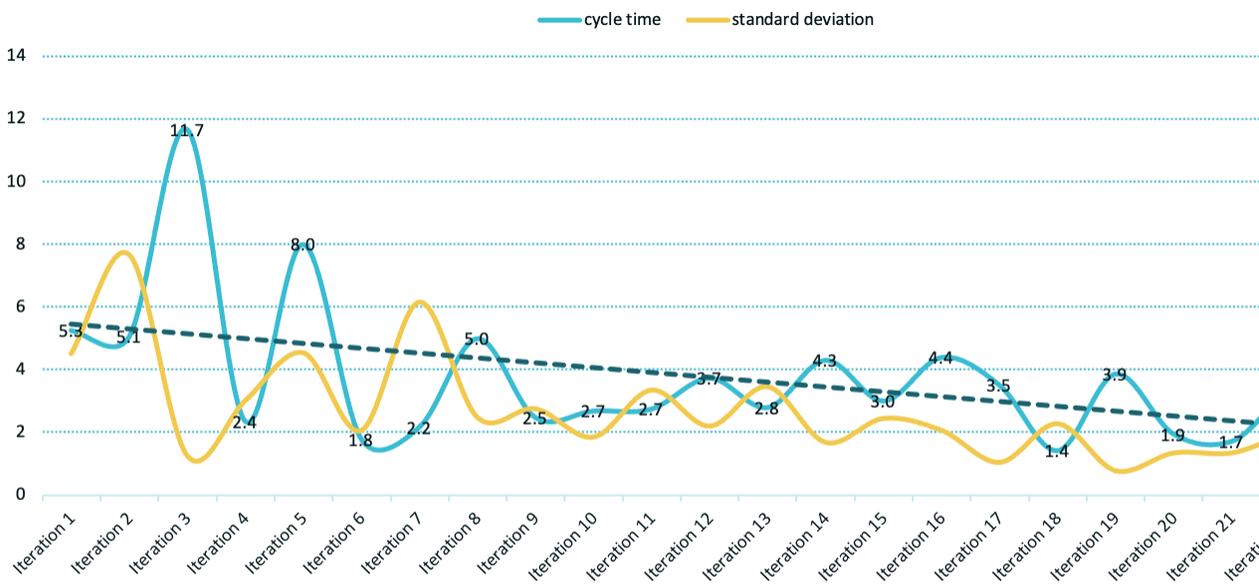
* 故事卡: 为客户提供价值的一小件任务/工作 * 吞吐量: 一个迭代(两周)完成的任务/故事卡总数 * 速率: 一个迭代故事总点数/实际人天

TOP1

主办方: msup

实践-协作维度-功能交付时长/阻塞时长

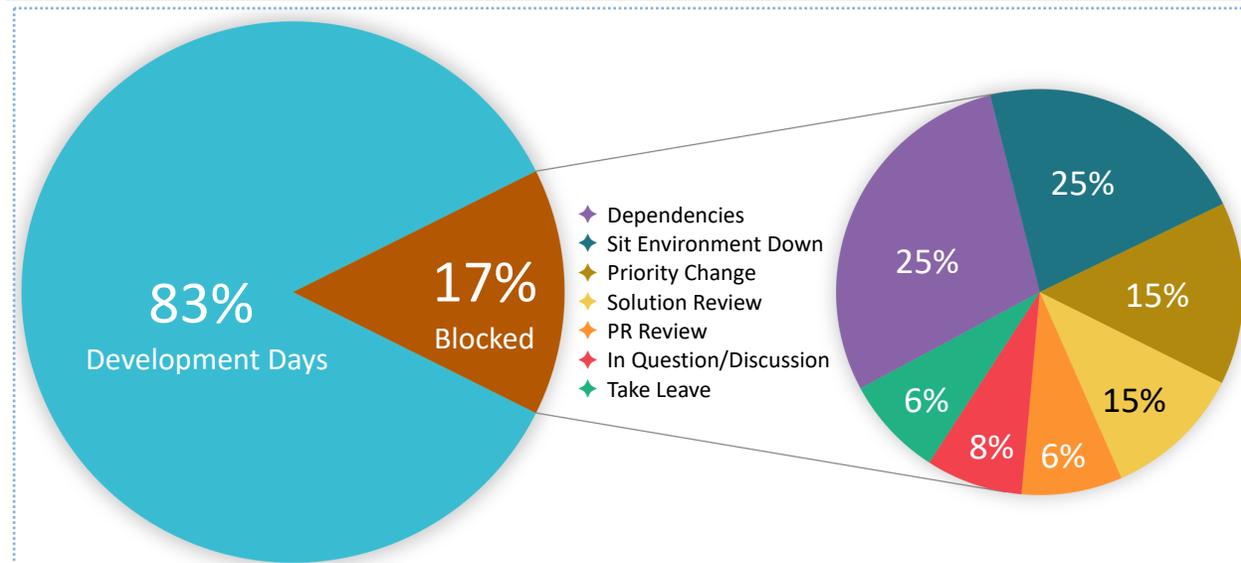
阻塞时长 - 每个迭代阻塞时间细分



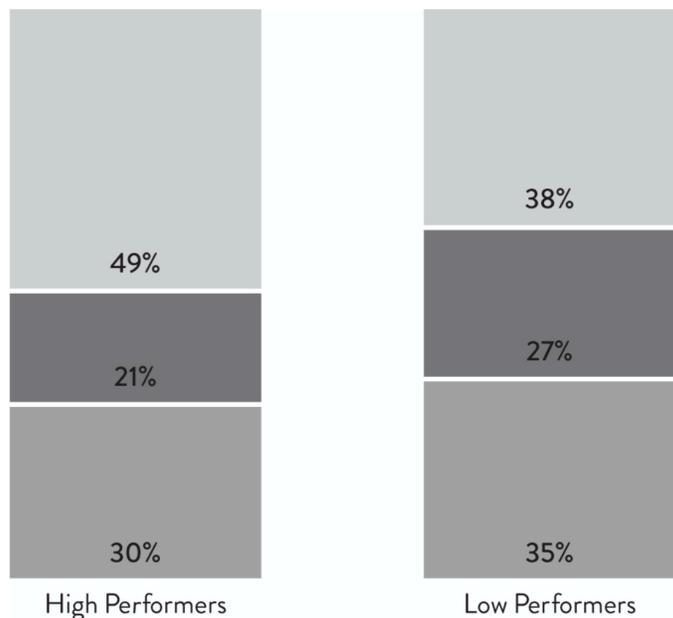
每个迭代平均功能交付时长(人天)和标准差

- ★ 你是否连续跟踪过团队多个迭代的平均功能交付时长?
- ★ 他们在交付功能时有没有被阻塞过?
- ★ 他们被阻塞了多久? 因为什么?

* 功能交付时长: 团队构建并交付一个任务/故事卡到客户手中所需要的天数



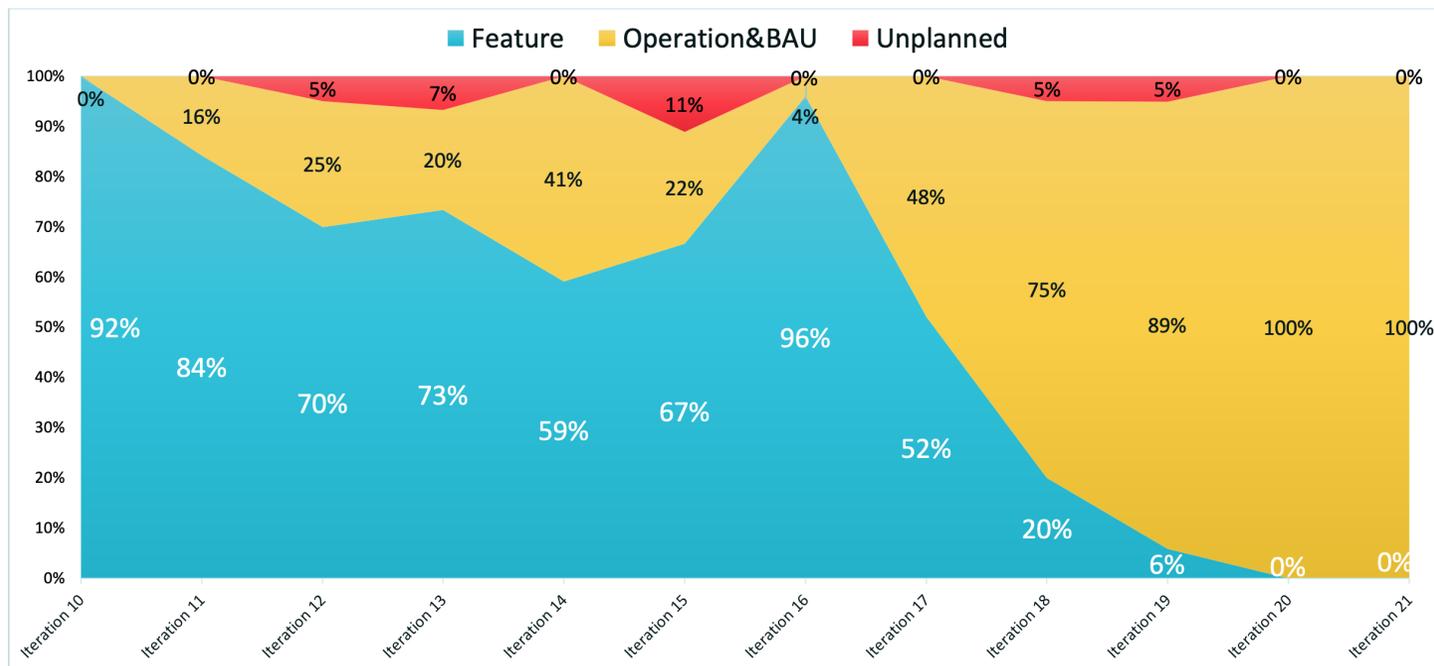
实践-协作维度-交付功能百分比



“High performers reported spending 49% of their time on new work and 21% on unplanned work or rework. In contrast, low performers spend 38% of their time on new work and 27% on unplanned work or rework.”

Nicole Forsgren PhD. “Accelerate.”

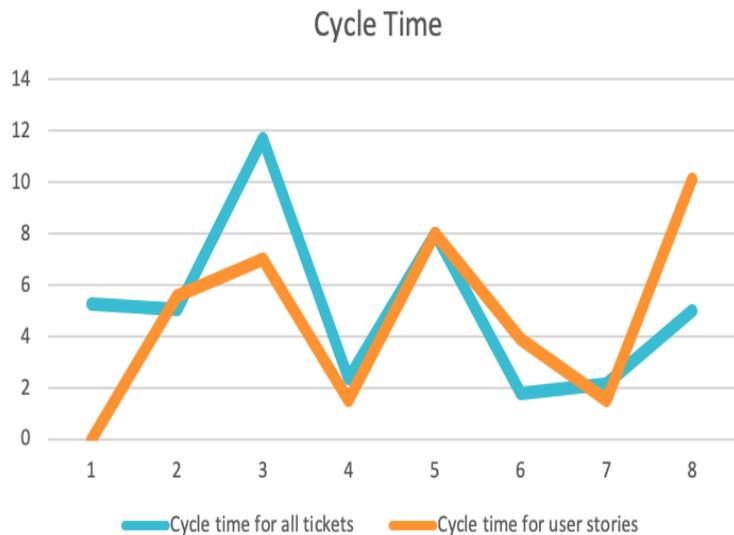
交付功能百分比 - 每个迭代交付功能百分比



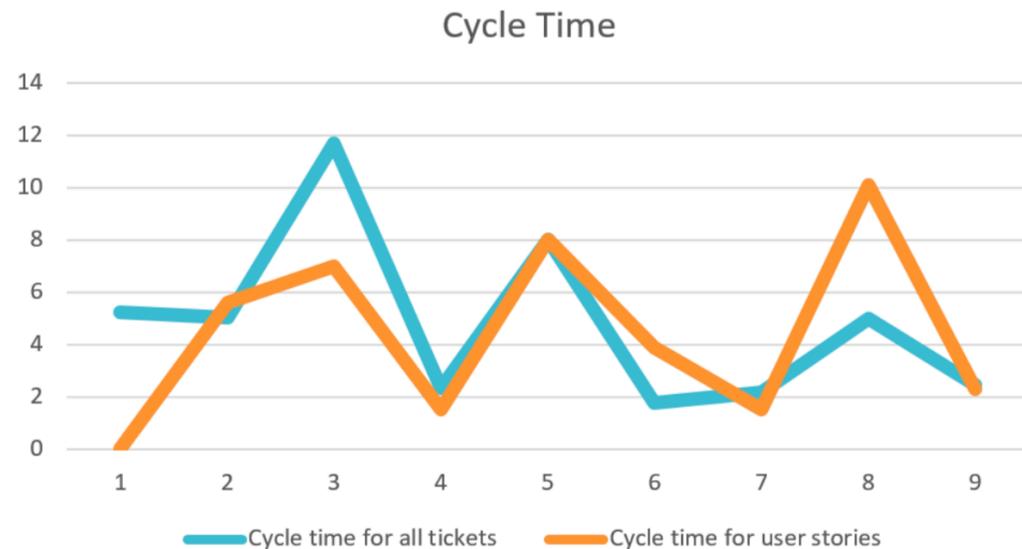
★ 你是否连续跟踪过团队多个迭代的交付功能百分比?

★ 是否团队交付的功能带来了业务上的正反馈?

实践-协作维度-功能交付时长-案例



迭代 8 → 迭代 9



★ 迭代 1 没有用户故事卡，所以数据是0。

★ Cycle time 在迭代 8 大幅上升，因为有一个大故事卡。(CT > 21天)

❖ 行动:

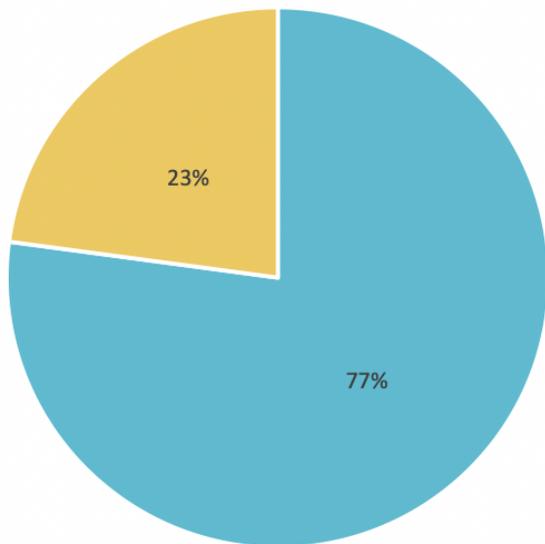
如果故事卡过大，团队将与业务分析师在工作开始时将其拆分成小故事卡。

★ 故事卡很快完成，因为它们被细分为小块。(行动所带来的好处)

★ 本迭代含有Bug修复和上线配置任务，这些任务完成的速度很快。

实践-协作维度-阻塞时长-案例

Developing VS Blocked



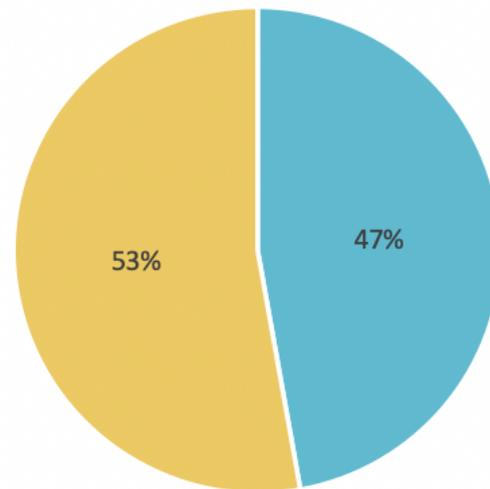
■ Total Development Time (DAYS): ■ Total waiting time (DAYS):

- ★ 被第三方系统 CRM 接口阻塞。
- ★ 强依赖设计师给出UI设计。
- ★ 团队被迫切换优先级，工作重排，切换和等待的额外耗时。

❖ 行动:

加速沟通频率或者重排任务优先级。
拉通相关部门/团队的OKR/KPI一致优先级。防止被迫切换。

Developing VS Blocked



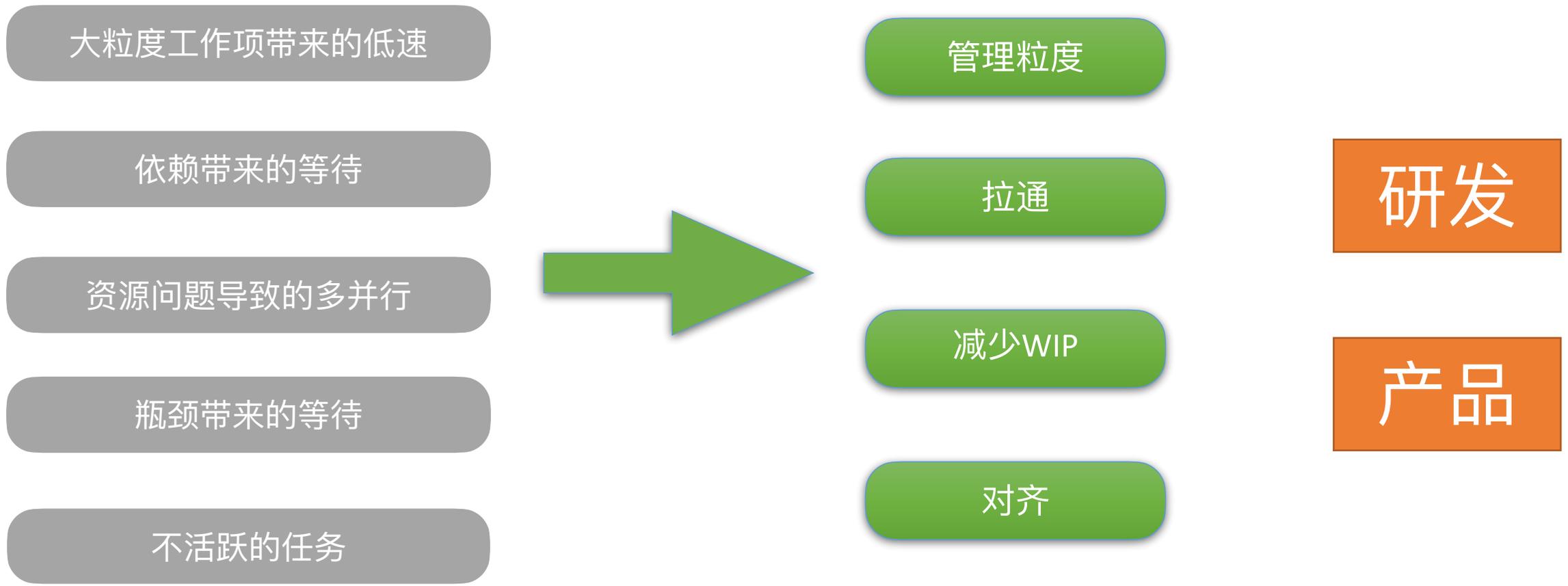
■ Total Development Time (DAYS): ■ Total block or waiting time (DAYS):

- ★ 第三方 CRM 阻塞更加严重，更多任务被阻塞。
- ★ UI设计基本明确，依赖基本解除。
- ★ 被迫切换的情况减少。

❖ 行动:

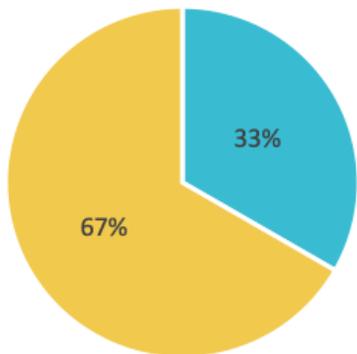
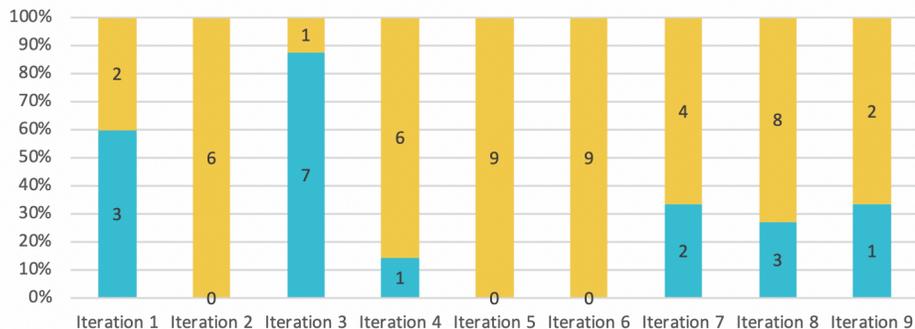
重排任务优先级，并上报产品和业务可能影响发布的风险。
持续拉通相关部门/团队的OKR/KPI一致优先级。

造成阻塞的原因和应对办法



实践-协作维度-交付功能百分比-案例

Proportion: User Story vs Non-Story



■ User story ■ Non User story

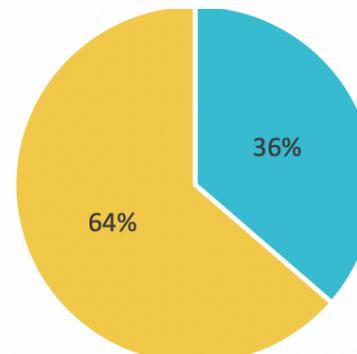
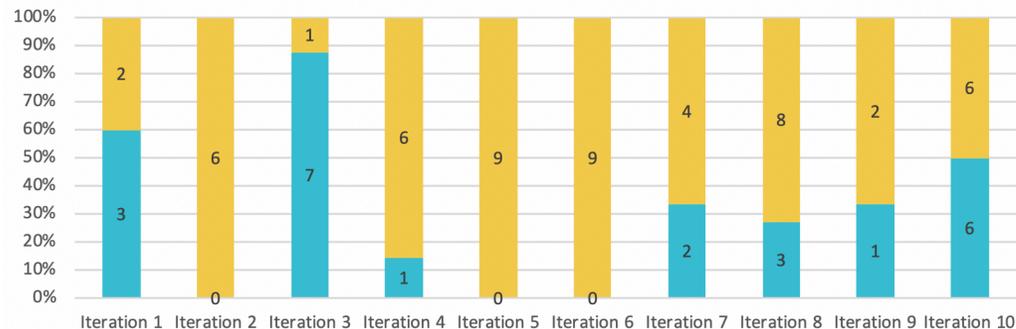
- ★ 实际开发需求并不明确，开发人员必须创建一些调研卡，以此明确故事卡试图实现的目标。
- ★ 交付了几个迭代，吞吐量不低，但是好像没什么业务回馈。

❖ 行动:

积极联系业务分析师，和他们结对创建故事卡。

反馈产品经理，当需求被验证并更加明确后，再进入开发工作。（需求有效性）

Proportion: User Story vs Non-Story



■ User story ■ Non User story

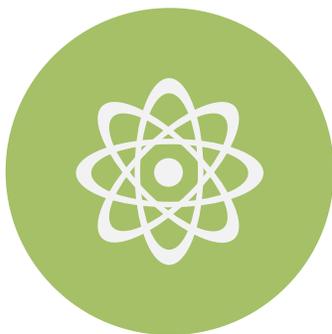
- ★ 更多的业务故事卡被创建。

❖ 行动:

继续和业务分析师结对创建故事卡。

迭代 9 → 迭代 10

100 实践-工程维度



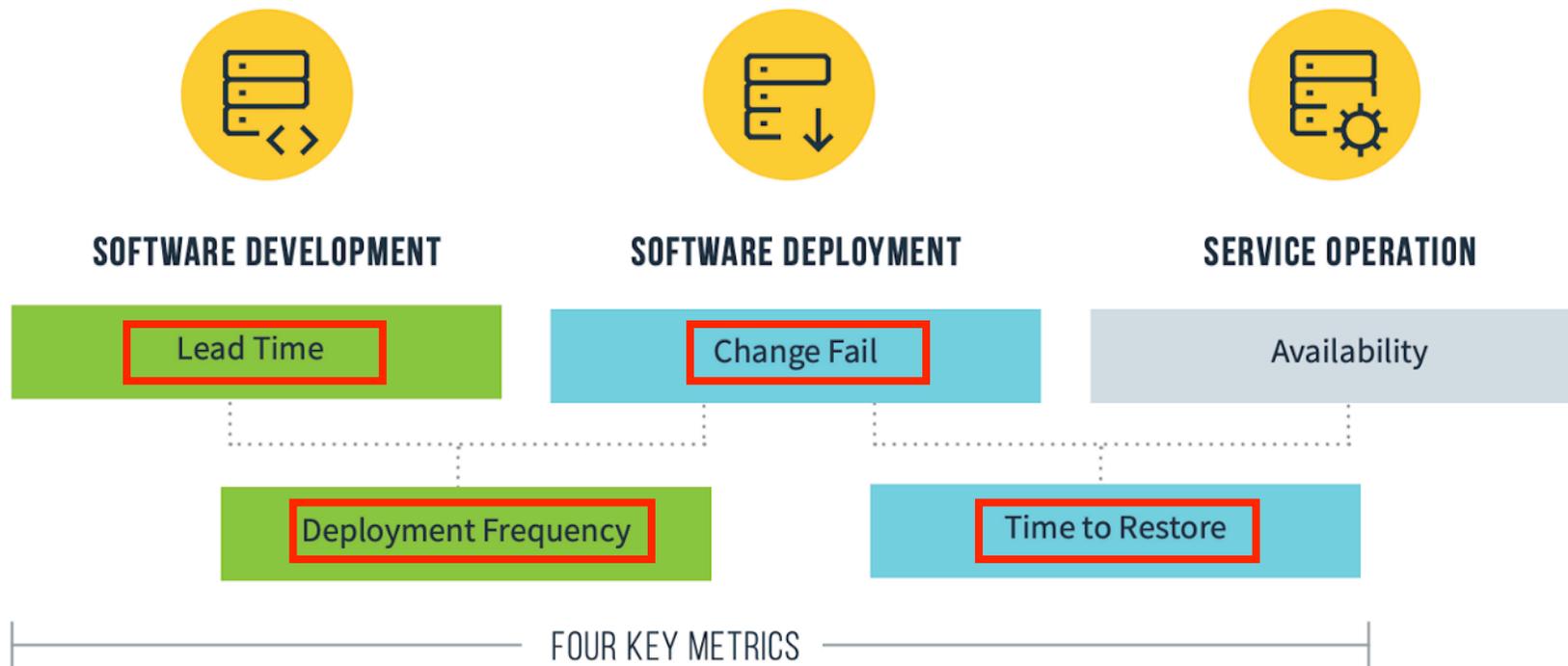
工程维度

变更前置时长

部署频率

上线失败率

平均恢复时长



实践-工程维度-思考

真正有用的指标是从目标出发的!

当你尝试制定一个指标的时候可以尝试先回答下面的问题：
(Goals-Questions-Metrics)

G

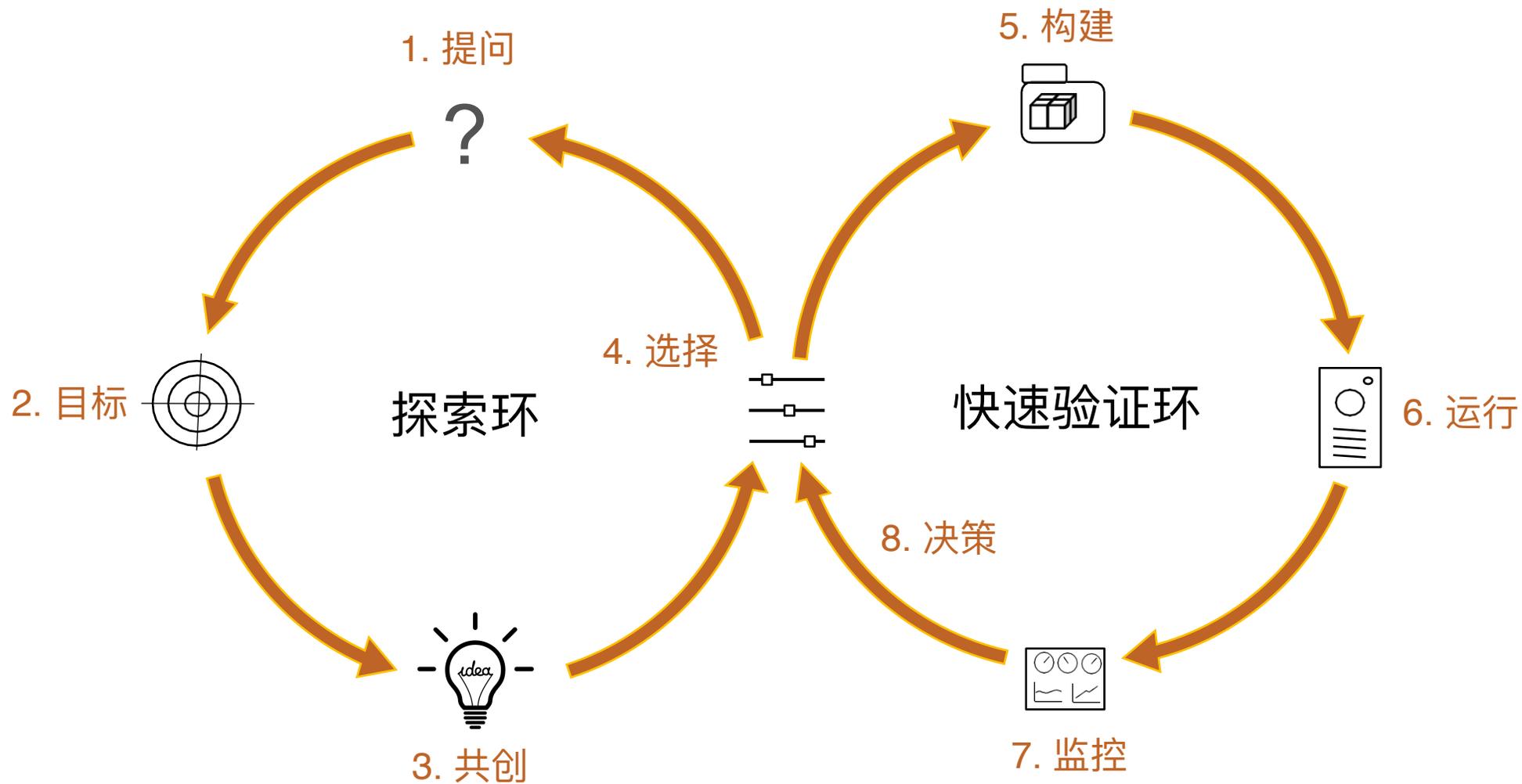
从这个指标中，我要解决的现实问题是什么？

Q

为了解决这些问题，我需要获得哪些相关信息？这些信息如何帮助解决上述问题？

M

使用了哪些指标，能最好的反应上述信息？

TOP100
在实践中找到合适自己的

* 乔梁《持续交付2.0：业务引领的DevOps精要》

实践-工程维度-思考

一些不够好的指标？不建议管理者设定与业务无关的效能指标



一些好的指标？（阿里211）

2周 交付周期：从想法提出并确认到上线的时间。

1周 开发周期：从需求设计完成（对开发就绪）到可上线的时间。

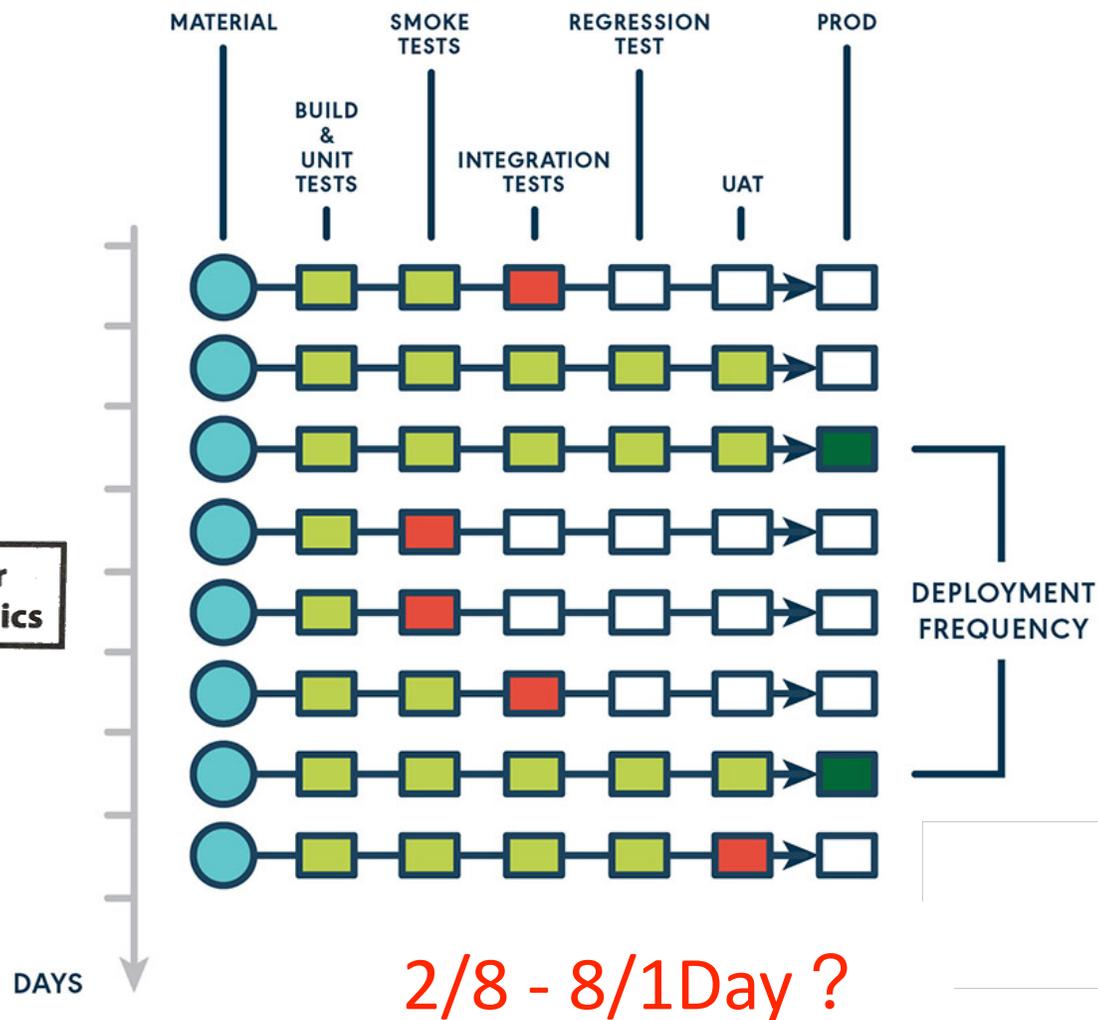
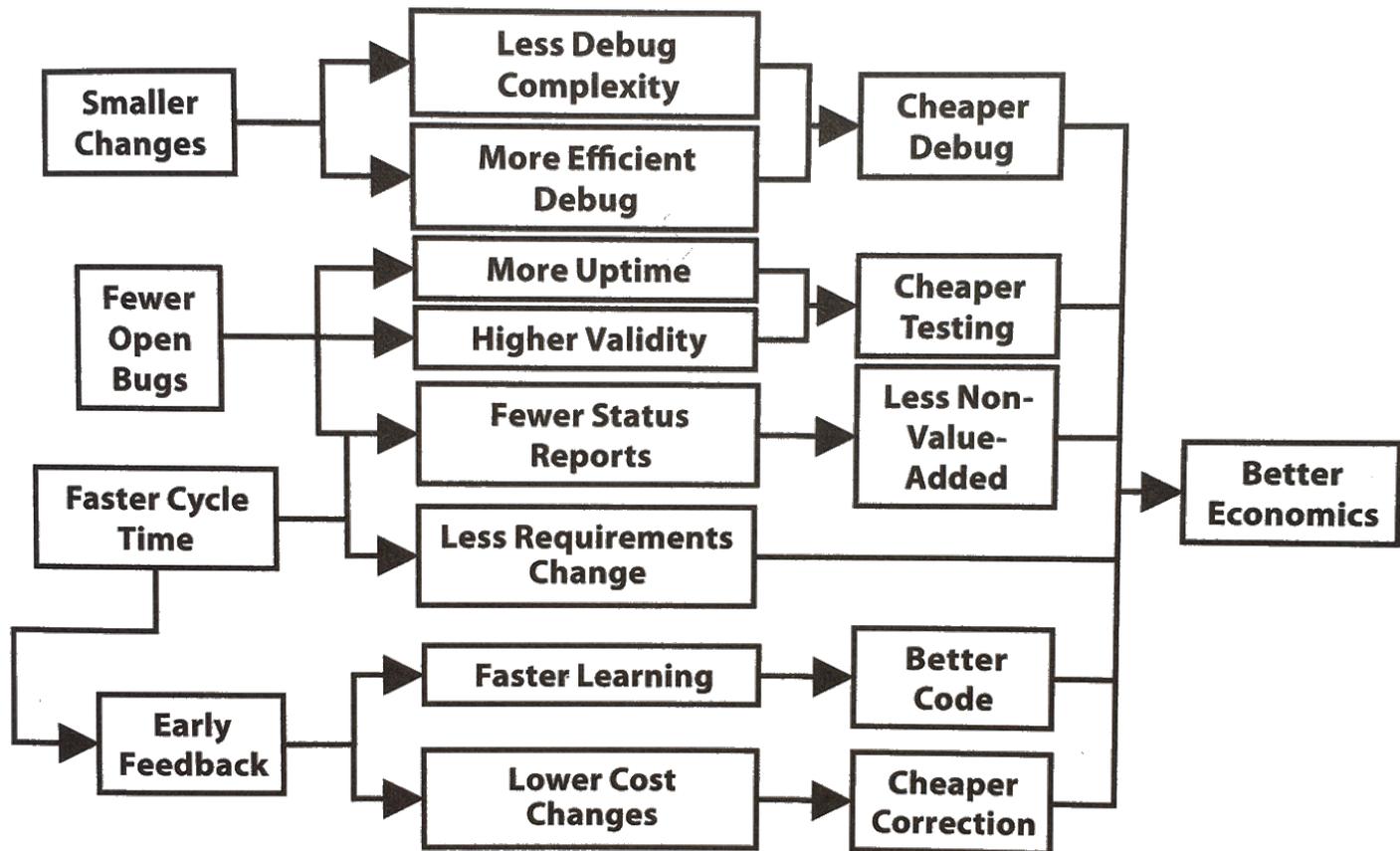
1小时 发布（变更）时间：代码完成后，上线所需要的时间。

古德哈特定律
当一个政策变成目标
它将不再是一个好的政策

结果指标是共识基础：需求有效性、交付效能、线上问题、研发成本。



实践-工程维度-部署频率

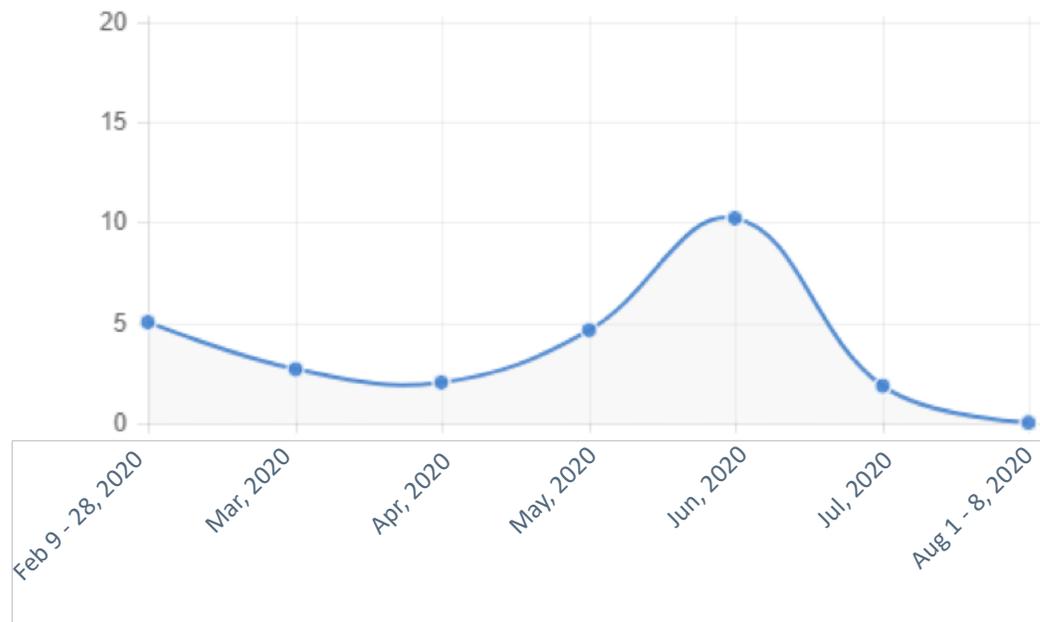


* 部署频率：部署到生产环境的频率

实践-工程维度-上线失败率-平均恢复时长-Failure Metrics

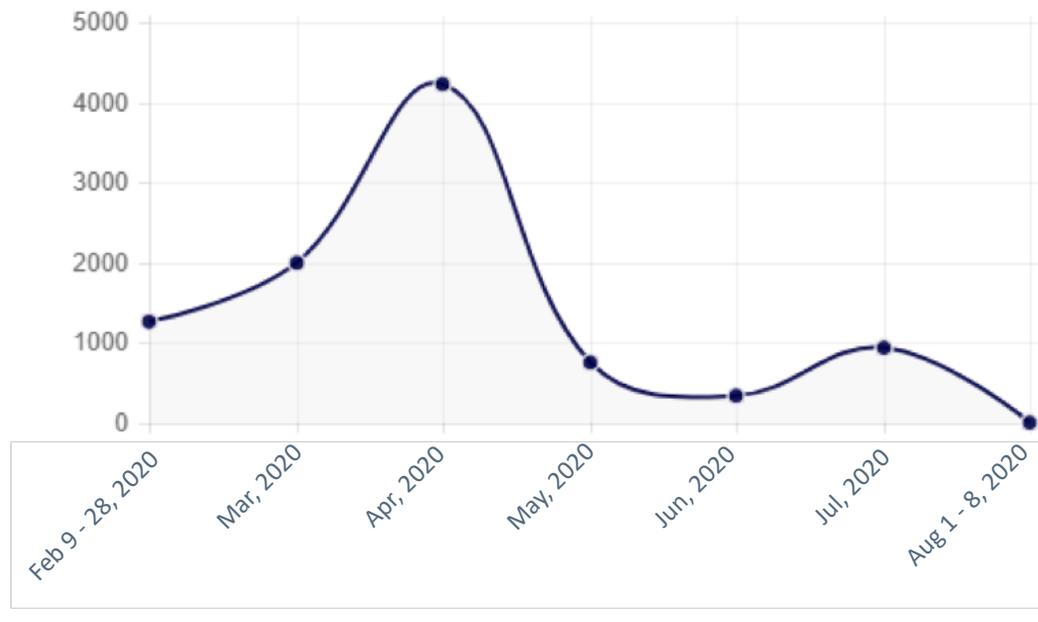
MTTR over time MTTR (平均维修时长)

Mean Time To Repairs - Feb 9, 2020 - Aug 8 2020



MTBF over time MTBF (平均无故障时长)

Mean Time Between Failure - Feb 9, 2020 - Aug 8 2020



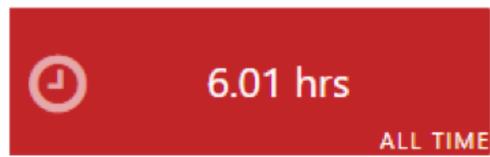
Total Downtime Hours



Total MTBF



Total MTTR



Total Time Spent



Custom Maintenance KPIs

* 上线失败率: 部署到生产环境后 (失败的次数 / 部署的次数)

* 平均恢复时长: 服务从部署失败到下次部署成功的修复时长总和/失败次数

实践-工程维度

	HIGH PERFORMERS	MEDIUM PERFORMERS	LOW PERFORMERS
Deployment frequency	On demand (multiple deploys per day)	Between once per week and once per month	Between once per week and once per month *
Lead time for changes	Less than one hour	Between once per week and once per month	Between once per week and once per month *
MTTR	Less than one hour	Less than one day	Between one day and one week
Change failure rate	0-15%	0-15%	31-45%

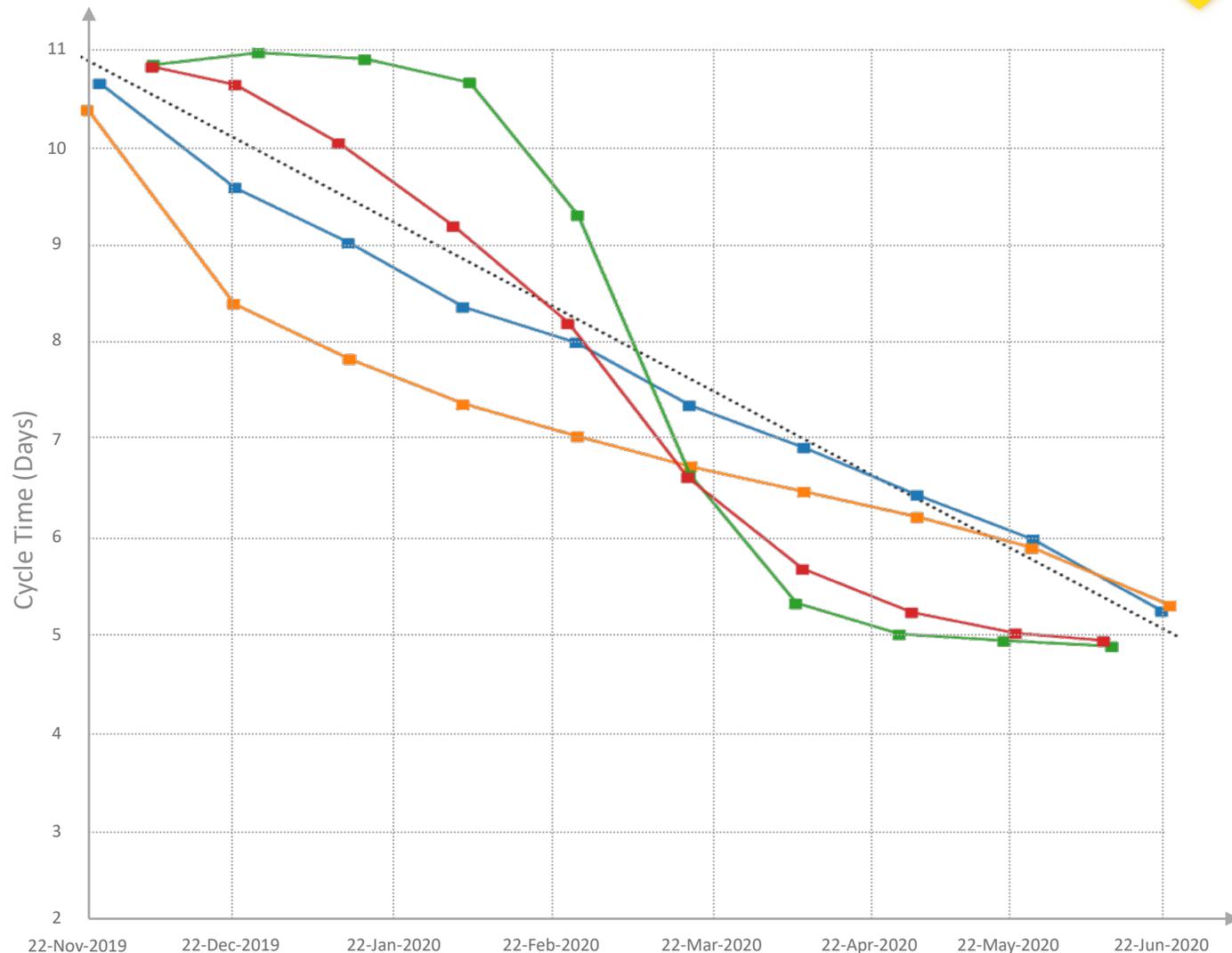
* Low performers were lower on average (at a statistically significant level) but had the same median as the medium performers.

案例ROI分析

多团队、多迭代、交付功能时长统计

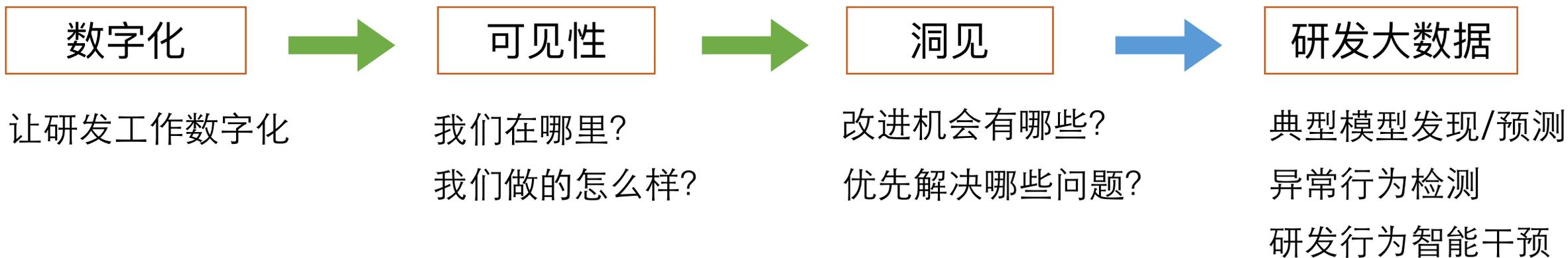
通过协作、工程维度的治理，交付时长明显缩短：
(8个月的连续治理)

- ★ 阻塞被有效识别并快速移除。
- ★ 平均每个任务/故事卡交付时长缩短了~0.5天。
- ★ 平均每个迭代/每个团队吞吐量上升了~4个。
- ★ 在当前项目规模上，每个迭代节省了50+人天。



案例启示

软件研发过程的数字化、可视化、让软件从业者终于有机会可以度量研发过程中各个阶段的效能。度量这些数字是让人兴奋的，终于有机会让它真正帮助到实践者。



案例启示

软件研发过程的数字化、可视化、让软件从业者终于有机会可以度量研发过程中各个阶段的效能。度量这些数字是让人兴奋的，但是如何有效的数字化、可视化、洞见让它真正帮助到实践者？



软件开发中没有银弹？

复杂性(Complexity)
 隐匿性(Invisibility)
 配合性(Conformity)
 易变性(Changeability)



我们所处的世界是VUCA的？

易变性(Volatility)
 不确定性(Uncertainly)
 复杂性(Complexity)
 模糊性(Ambiguity)



协作维度

速率/吞吐量
 功能交付时长/阻塞时长
 交付功能百分比

工程维度

变更前置时长
 部署频率
 上线失败率
 平均恢复时长

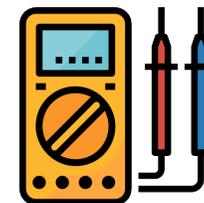
案例启示

软件研发过程的数字化、可视化、让软件从业者终于有机会可以度量研发过程中各个阶段的效能。度量这些数字是让人兴奋的，但是如何有效的数字化、可视化、洞见让它真正帮助到实践者？

软件开发中没有银弹？我们所处的世界是VUCA的？

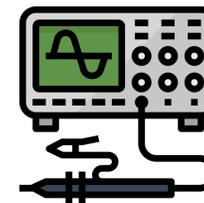
当你从业务价值角度发现交付团队响应客户、产品团队需求慢的时候？

协作维度：速率/吞吐量、功能交付时长/阻塞时长、交付功能百分比



当你需要更细粒度的从工程层面了解阻塞发生在哪里时？

工程维度：变更前置时长、部署频率、上线失败率、平均恢复时长



101 下一步启示

- ★ 研发活动的全面数字化，带来了研发效能可见、可改进的巨大空间。
- ★ 研发效能度量被寄予厚望，现实仍有不足。
- ★ 面向结果设计可见性指标，面向改进设计过程性指标。
- ★ 通过协作维度、工程维度，识别效能改进的机会点。
- ★ 所有改进需要回馈业务成果。



微信官方公众号：壹佰案例
关注查看更多年度实践案例

附录

标准差（又称标准偏差、均方差，英语：**Standard Deviation**，缩写**SD**），数学符号 σ （sigma），在**概率统计**中最常使用作为测量一组数值的**离散程度**之用。标准差定义：为**方差开算术平方根**，反映组内个体间的离散程度；标准差与**期望值**之比为**标准离差率**。测量到分布程度的结果，原则上具有两种性质：

1. 为非负数值（因为开平方后再做平方根）。
2. 与测量资料具有相同单位（这样才能比对）。

简单来说，标准差是一组数值自**平均值**分散开来的程度的一种测量观念。一个较大的标准差，代表大部分的数值和其平均值之间差异较大；一个较小的标准差，代表这些数值较接近平均值。

例如，两组数的**集合**{0, 5, 9, 14}和{5, 6, 8, 9}其平均值都是7，但第二个集合具有较小的标准差。

例如，A、B两组各有6位学生参加同一次语文测验，A组的分数为95、85、75、65、55、45，B组的分数为73、72、71、69、68、67。这两组的平均数都是70，但A组的标准差约为17.08分，B组的标准差约为2.16分，说明A组学生之间的差距要比B组学生之间的差距大得多。

基本定义

$$SD = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2}$$

DELIVERY METRICS

HEARTBEAT

HeartBeat is a tool for tracking project delivery metrics that can help you get a better understanding of delivery performance.

This product allows you easily get all aspects of source data faster and more accurate to analyze team delivery performance which enables delivery teams and team leaders focusing on driving continuous improvement and enhancing team productivity and efficiency.

<https://github.com/thoughtworks/HeartBeat>

4 hours to 4 minutes



Import Project From File

Create A New Project

