

## 未来工厂 - 场景 #235

### 元宇宙解决下一个关键矛盾就是算力规模超过人应掌握合理知识的极限

十二月 30, 2022 03:31 下午 - mia

状态:	Storytelling:Automation	开始日期:	十二月 30, 2022
优先级:	普通	计划完成日期:	
指派给:		% 完成:	60%
类别:		预期时间:	0.00 小时
目标版本:	智能制造规划师		
关联联系人:			
<b>描述</b>			
<p>如果说历史上的第一（结构化编程扭转了当时直接面向全局数据、满屏 Goto 语句书写面条式代码（Spaghetti Code）的编程风气，强调可独立编写、可重复利用的子过程/局部块的重要性，让软件的每个局部都能够设计专门的算法和数据结构，允许每一位程序员只关注自己所负责的部分，从而在整体上控制住了软件开发的复杂度。此时，软件的架构才开始独立于硬件物理架构而存在，软件业开始出现把控全局设计的架构师与聚焦局部实现的程序员的角色分工。））、二次(面向对象编程逐步取代了面向过程的结构化编程，成为主流的程序设计思想。)软件危机分别是机器算力规模超过了人类个体的生理极限，超过了人类群体的沟通极限的话。那么在今天，在云计算的时代，数据中心所能提供的算力其实已经逼近人类协作的工程极限。与此算力相符的程序规模，几乎也到了无论采用何种工程措施去优化过程、无论采用什么管理手段去提升质量，都仍然不可避免会出现意外与异常的程度。</p> <p>我认为，软件发展的下一个关键矛盾将会是算力规模超过人应掌握合理知识的极限。经过良好设计的分布式系统，拥有局部的可再生性，确实能在整体上展现出可靠的服务能力。然而，“良好地设计”一个分布式系统很不容易。今天无限火热的云原生、微服务、不可变基础设施、弹性计算、服务网格、无服务器架构、高低零代码等等，背后都能展开成一整套成体系的开发或者设计方法。这些新的技术在为人们解决了更复杂软件问题的同时，也正在把编程这件事情本身的复杂度推向更高层次。</p>			
<b>相关的问题:</b>			
关联到 未来工厂 - 场景 #227: 未来工厂能力元		Verify:需求验证	十二月 29, 2022
关联到 未来工厂 - 场景 #205: 为什么数字化企业是能力驱动而不是面向过程的？		Develop:需求分发	十二月 07, 2022
关联到 未来工厂 - 场景 #234: 从智能制造发展历程看智能制造的推进		Define:需求分析	十二月 30, 2022

#### 历史记录

#2 - 十二月 30, 2022 03:32 下午 - mia

- 关联到 场景#227: 未来工厂能力元 已添加

#3 - 十二月 30, 2022 03:32 下午 - mia

- 关联到 场景#205: 为什么数字化企业是能力驱动而不是面向过程的？ 已添加

#4 - 十二月 30, 2022 03:42 下午 - mia

- 状态 从 Define:需求分析 变更为 Storytelling:Automation

- % 完成 从 20 变更为 60

#5 - 七月 24, 2023 10:46 上午 - rick

- 关联到 场景#234: 从智能制造发展历程看智能制造的推进 已添加

#6 - 十一月 10, 2023 12:15 下午 - rick

- 目标版本 从 未来工厂转型 变更为 智能制造规划师

#7 - 三月 24, 2024 05:40 下午 - rick

- 标签 从 meta, futurefactory 变更为 meta, futurefactory, im