

基于 Scrum 方法的“在线考试支持系统”

软件过程分析、定义及实施

南京大学软件学院

151250048

郭浩滨

项目概况

当下校园内的一个重要活动——考试，尤其是电子/在线考试遇到了诸如考试系统不完善、性能差、无法稳定支持较多人数同时考试的一系列问题。在这个背景下，在线考试支持系统这个市场有很好的前景，于是我们打算开发一个基于 web 的在线考试支持系统。该系统具有稳定、快速等特点，最多支持 200 人同时参加考试。经讨论，该在线考试支持系统的需求规格说明已经基本完成，系统需要提供邮件提醒、考试管理、考试过程支持等功能。开发周期为 4 周或 4 周以内。

项目分析及软件过程设计

参考《在线考试支持系统需求描述》，我们一共需要完成邮件提醒和以课程为单位的考试过程管理这两个核心功能，同时支持快速、稳定、支持并发等质量属性。分析该需求规格说明我们可以得知已经有较为成熟的技术来实现该系统，如使用基于 Spring Cloud 的 web 开发方案。再考虑项目的时间限制，我们只有小于 4 周的时间实现该系统（最好为 3 周~3 周半）。综上，我们可以采用基于 Scrum 方法的敏捷开发过程。

Scrum 是目前当红的敏捷开发方法，它关键的目的和特点是适用变化和快速迭代。它的一个基本假设是外部需求模糊而难以理解。而这也契合在线考试支持系统现阶段的情况，由于该系统现有参考案例不足，需求难免会产生变更，如导入的格式由 excel 模板临时变更为 csv 等等。严格控制变更的同时敢于承认变更的存在并迅速反应，是 Scrum 方法适合这个系统的重要原因。同时，我们只有最多 4 周的可控时间来完成这

个系统的开发，所以相对于瀑布模型的高度文档化与规范化，快速迭代是我们重要考虑的软件过程目标。同时，由于该系统 4 周后就需要正式投入使用，所以在开发过程中需要严格控制软件质量及做好风险预估和防范。

实施 Scrum 方法之前，我们必须做好以下的准备工作以及明确相关的概念和目标：

团队角色分配

根据以上的论述，结合 Scrum 方法特点，预估团队应由以下角色组成：

角色分工	主要描述
Product Owner	产品负责人，确定“大家要做什么”。充当项目的主要利益相关者和维护者，主要工作有维护产品待办事项列表，与其他利益相关者沟通，决定项目需求优先级等等。可以从我们的团队中挑选一名有产品负责经验的人员担任，尤其是做过产品经理工作的人员。
Scrum Master	Scrum 的推动者，充当团队的“教练”。主要负责确保团队按时按质完成迭代目标，组织会议，优化工作流，激励开发团队等等。对于这个项目而言，最好挑选有 Scrum Master 经历的人员当任，或者有相关开发教练工作或管理工作的人员。
Team	一般由多个开发者组成，是开发的主力。在这个项目中，开发人员的数量规模应在 7~9 个人左右。如 3 个成员主攻前端开发，4 个成员主攻后端等等（但知识体系应有所交叉）。

角色分配过程中，我们要注意以下一些问题：

- 正确定位队伍中的 Scrum Master：Scrum Master 应确保流程执行正确，避免因对 Scrum 过程不够了解和适应而导致将短时间的瀑布周期与真正的 Scrum 迭代相混淆的情况，比如有意无意地拉长 Sprint 周期，并严格区分计划周、开发周、测试周等做法。在这种具体的项目中，Sprint 周期最好定义为一周（工作日）。
- 构建交叉的知识架构：组建过后的整个开发团队每个人的知识体系应该是各有专长但又互有交叉的。团队中不应该发生“开发的只管开发，测试的只管测试”的情况。团队中的开发成员应尽可能地知道项目的各种细节，以应对临时人员不足

（如请假等情况）导致的项目停滞。这要求团队在每日站会和日常开发中需要多一些高效的交流和问题探讨。

Scrum 工具选择

Scrum 的核心要素是产品订单（Product backlog）、冲刺订单（Sprint backlog）和燃尽图（Burn down chart）。产品订单是整个专案的概要文档。

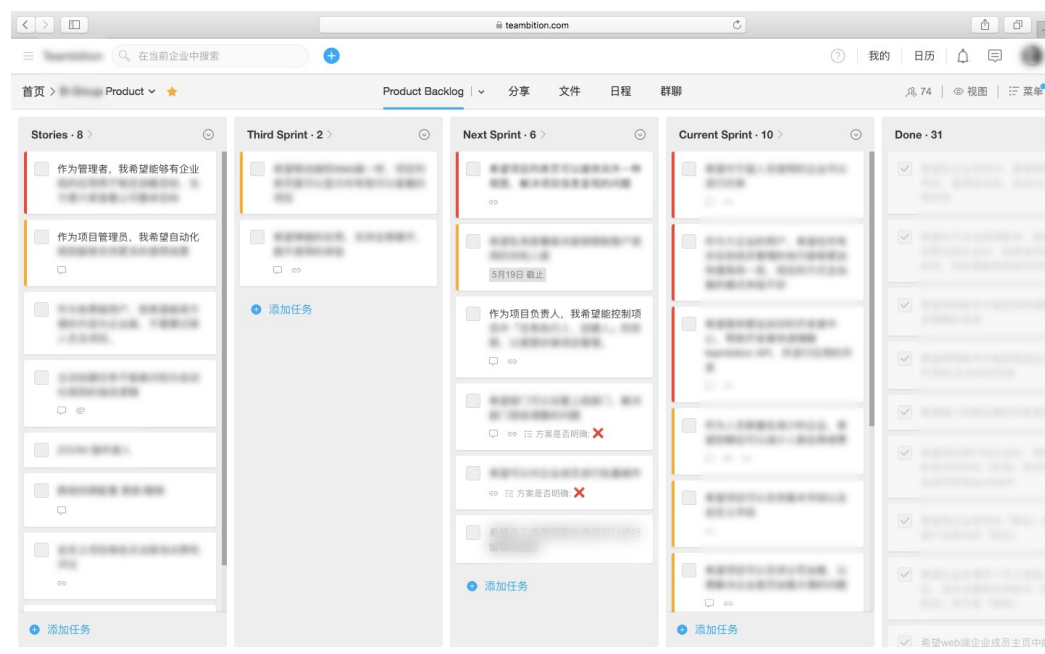
- 产品订单包括所有所需特性的粗略的描述。产品订单是关于将要生产什么样的产品。产品订单是开放的，每个人都可以编辑。产品订单包括粗略的估算，通常以天为单位。估算将帮助产品负责人衡量时程表和优先级（例如，如果"增加拼写检查"特性的估计需要花 3 天或 3 个月，将影响产品负责人对该特性的渴望）。
- 冲刺订单是大大细化了的文档，包含团队如何实现下一个冲刺的需求的信息。任务被分解为以小时为单位，没有任务可以超过 16 个小时。如果一个任务超过 16 个小时，那么它就应该被进一步分解。冲刺订单上的任务不会被分派，而是由团队成员签名认领他们喜爱的任务。
- 燃尽图是一个公开展示的图表，显示当前冲刺中未完成任务数目，或在冲刺订单上未完成的订单项的数目。燃尽图可能在一次冲刺的大部分时间内都维持平坦，但计划仍然可以按照既定时间进行。

传统的 Scrum 会用白板和 Excel 等工具来实现上述三种核心实践，这些传统方法相对来说效率较低，且隐性地增加了时间空间等限制，不够现代化、自动化。因此，我们可以使用相对较为流行的现代化协作工具，如 Teambition。

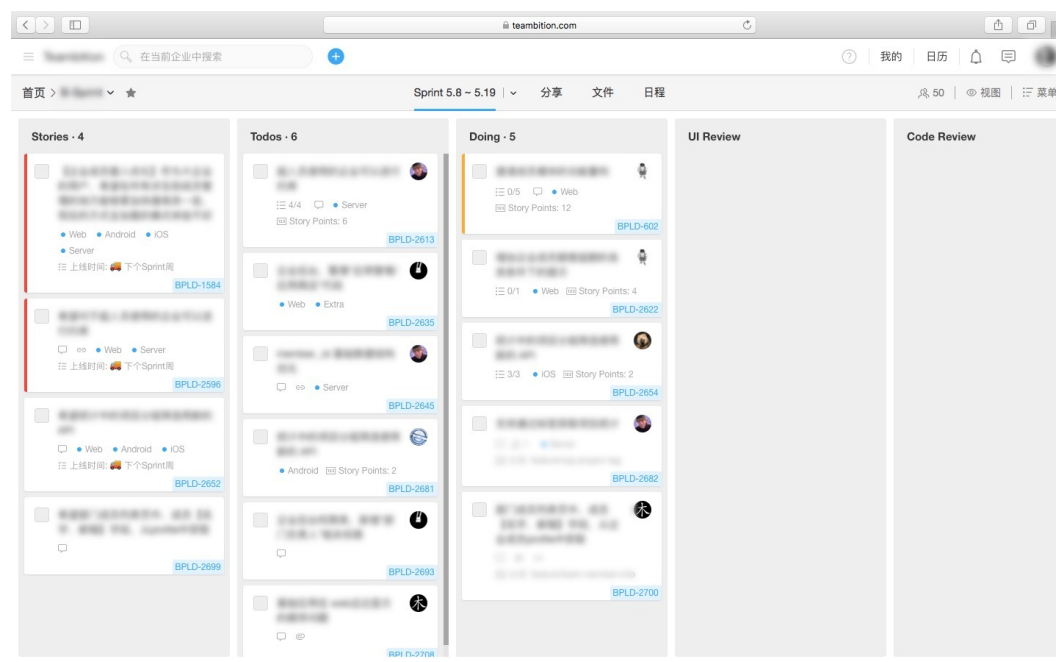
Teambition 是敏捷团队协作的一站式解决方案，它通过帮助团队轻松共享和讨论工作中的任务、文件、分享、日程等内容，极大的方便了团队间的各种沟通形式，并在网页、桌面、移动环境都打造了体验出众的应用，满足了“随时随地”的要求。更可贵的是，Teambition 有针对 Scrum 团队的一套经优化的自动化流程，对 Backlog 的管理和会议活动管理非常方便，能更好地实现团队协作、需求到研发的信息关联与共享。比如 Teambition 中的「任务看板」功能，把所有任务及进程都平铺开来说显示，实现

了便捷的项目流程化管理；「分享」功能，大家在其中共享工作信息、总结知识经验；「文档」相当于该项目的共享协作网盘，大家能随时访问、更新共享资料；「群聊」则更像是该项目的群聊，支持项目成员随时沟通想法，并且自动保留每一条消息。

如 Product backlog:



Sprint backlog:



还可以实现不同 Backlog 间的联动，如从 Product backlog 中提取需求到 Sprint backlog。

质量控制及风险预估

软件的质量不仅仅取决于软件的稳定性、性能等质量属性，还取决于软件是否符合用户的预期，满足用户的真正需求。所以在构造用户故事点的时候，应该讨论得出准确的原型，尽可能确保最终产品是满足用户需要的。在开发布计划会议时，也应对文档和冲刺目标进行 Review。同时，团队个人和小组应该及时做好 Code Review 和 Unit Test 等工作，检查组件架构、层级结构是否清晰，代码是否冗余、符合代码规范，单元测试覆盖率是否满足预先要求等等。

过程实施

总体时间安排

时间安排	主要活动
第 1 天	项目启动会议和需求评审会议。上午开始项目启动会议，主要确定团队的人员安排，项目背景和大方向，技术栈讨论，技术人员调整，确定总体方向、团队纪律、约定等内容。下午进行需求评审会议，对已有的文档产品——《在线考试支持系统需求描述》进行评审，确保其满足一下五个特点——正确性、完整性、一致性、简洁性、规范化，并按照评审微调后的文档内容确定产品待办事项及其优先级（由 Product Owner 最终确认）等。
第一周（2-5/6 天）	正式进行第一个 Scrum 冲刺。在冲刺的开始，需要进行一次两小时左右（第一次可以延长最多一个小时）的 Sprint 计划会。会议主要内容是 Project Owner 向大家介绍排好序的 Product backlog，然后大家共同思考决定如何推进计划，梳理出 Sprint backlog 并预估每个 backlog 的规模/预计完成时间（可以用计划卡片等工具）来完成后续的工作。随后正式进行冲刺。在冲刺中，需要进行每日站会，主要汇报三点：1. 从昨天 Daily Scrum 到这一刻，我完成了什么工作？2. 从这一刻到明天的 Daily Scrum，我计划完成什么工作？3. 是否有什么困难阻碍了我的进展？预计时间为 15 分钟。在

	Sprint 结束后，需要进行 Sprint 评审会。主要评审本次 Sprint 的产出，并根据实际情况，适度调整 Product backlog。最后进行 Sprint 回顾会议，讨论如何优化 Sprint 工作流程。
第二周	与第一周相似，从剩下的 Product backlog 中适量选出生成 Sprint backlog 并进行冲刺。不同的点在于，可以根据第一周 Sprint backlog 的完成情况对各个 task 的规模进行更精确的估算。
第三周	同第二周。至此项目应该集成完毕，等待最终的系统测试。

开发过程详细计划

在第一周项目应根据项目启动会议确定的技术栈（如使用 Spring Cloud 或 PHP 技术）搭建该 web 项目的整体框架，并根据 Sprint 计划会议上确定的 task 列表逐步进行集成，第一周的冲刺实现的 Product backlog 尽量保证核心、相对容易实现，比如导入试题等，要求每一次代码集成都应该编写单元测试，保证较高的测试覆盖率。在相对比较难处理的问题逻辑或代码实现上，可以考虑实践结对编程，保证代码质量。在第一周的最后一天进行 Sprint 评审会议，分析燃尽图，找出 Scrum 实践中遇到的种种问题。

第二周应重点攻克项目中逻辑较为复杂的部分，如成绩计算、生成试题等，并优化 Sprint 过程。

第三周可以着重考虑项目的性能表现，在测试时采用 JMeter 等工具对软件进行压力测试，确保其满足性能目标。

优秀实践

- 持续集成、交付和部署

持续集成是敏捷方法中的一个优秀实践，开发者可以使用 Jenkins 等持续集成工具对软件制品进行持续集成。开发人员提交了新代码之后，立刻进行构建、运行（单元）测试。根据测试结果，我们可以确定新代码和原有代码能否正确地集成在一起。

- 测试驱动开发（TDD）

- 结对编程
- 自动化回归测试
-

测试、发布及培训

在最后一周的前 1-3 天，对系统进行完整的 alpha 和 beta 测试，并应对系统潜在的 bug，验证其是否真正满足客户的需求。注意同时要在目标场景，即真正考试环境下测试。在测试中不仅要软件进行可用性、稳定性、人性化等进行全方位测试，还应该模拟 200 人同时考试的高压环境，测试系统的并发性能。第 4-5 天，发布产品，编写产品使用指南文档并对客户/系统使用人员进行系统化的培训。