软件过程改进期末复习（TSPi）

为什么学习 TSPi？

它指导学生进行有效的团队合作和过程方法

第一章 TSPi 回顾

周期性开发策略：从最小的可用版本开始进行。

周期性开发策略约束：

1.每个周期都应该是可以测试的版本；

2.每个周期应该足够小以便在可用时间中进行开发和测试；

3.集成时，可以得到期望的产品。

第二章 团队软件过程的逻辑

一般的团队问题

1.领导效率低

2.合作失败

3.缺乏参与

4.拖延和缺乏自信

5.低质量

6.功能蔓延

7.低效率的评估

什么是团队？

1.最少有两个人，who

2.一起为一个共同目标而工作

3.每个人被分配了具体的工作或功能

4.工作的完成需要组员中的表格依赖

建立一个高效的团队

1.Team Cohesion	团队凝聚力
2.Challenging Goals	挑战的目标
3.Feedback	反馈
4.Common Working Framework	共同的工作框架

TSPi 如何构建团队（GRE PC）

1.Goals

2.Roles

3.Plans

4.Communication

5.External Communication

第三章 运行一个团队项目

团队角色

1.Team Leader

2.Development Manager

3.Planning Manager

4.Quality/Process Manager

5.Support Manager

为什么要有团队目标？

团队目标决定了框架和策略

TSPi 的目标建立

1.Team Goal1：Produce a quality product.

a) 第一次编译之前找出缺陷百分比：80%

b) 系统测试发现错误数：0

c) 工程完成时包含需求功能：100%

2.Team Goal2：Run a productive and well-managed project.

a) 估计产品规模错误：<20%

b) 估计开发时间错误：<20%

c) 记录和进入工程记事本数据的百分比：100%

3.Team Goal3：Finish on time.

a) 开发周期结束时间提前或结束：<4

团队角色分工

1.Team Leader

(1)组建和维护高效的团队

(2)增强团队成员的工作激情

(3)解决队员带来的问题

(4)向 Instructor 汇报团队进度

(5)组织团队会议

2.Development Manager

(1)生产产品

(2)根据成员的能力和技能分配任务

3.Planning Manager

(1)制定详尽、精细的计划

(2)每周准确的汇报团队状况

4.Quality/Process Manager

(1)准确汇报队员情况、正确使用 TSPi 进程数据

(2)保证团队遵照 TSPi 开发产品

(3)适度检查并报告

(4)会议记录，并写入工程记事本

5.Support Manager

(1)保证合适的开发工具和方法

(2)没有不被授权的改变

(3)风险记录系统记录所有风险和事件

(4)在团队开发周期中达到重用目标

团队会议：一起分析团队上周数据和开发周期日期

第一次团队会议：队长召开新团队的第一次会议

1.讨论团队成员角色

2.复查并更新团队目标

3.决定一个每周例会的标准时间

4.讨论出一个团队成员向计划经理提交每周数据的具体时间

第四章 开发策略

什么是策略？

策略就是找出如何构建一个大系统。

策略的目标：最小化风险

TSPi 的基本策略：在周期性进程中开发产品

策略如何进行？

1.定义主要策略

2.决定可选策略

3.定义可选策略的风险和利益

4.评估这些风险

5.做出策略性决定

6.文档化选择的策略

制定一个概要设计

1.概要设计应该包括在开发周期中准备实现的所有功能

2.作为策略进程的一部分，需要决定每个周期开发什么功能

制定初步规模估计

1.假定功能和部件之后，考虑每个程序元素，判定有多少新加和改变的 LOC。

2.将这些估计数据写入 STRAT 表

3.使用估计的程序规模和一个 LOC/hour 比率，估计开发每个功能可能花费的时间

4.计划经理带领团队制定一个初步的规模和时间估计。

什么是风险？

风险就是可能或可能不发生的事件。

SCM：从开发进程开始到结束，用来控制软件产品目录的一系列活动

SCM 必须包括：CIP，CCP，CCB

CIP（配置定义计划）的目标是保证

1.产品命名唯一

2.产品基线完成点

3.产品拥有者定义

CCP（配置控制规程）的目标时保证

1.同一时间不允许多个工程师对同一产品进行修订

2.修改是被授权的

CCB（配置控制委员会）：保证基线产品不能被随意修改，所有的修改要提交表格给 CCB 获得授权。

第五章 开发计划

根据计划跟踪进程

PV（计划价值） 计划部分时间/计划总时间×100%

EV（获得价值） 实际部分时间/实际总时间×100%

对工程师个人而言，TASK 的粒度应该小于 10 小时

计划中要预留 5%——10%的杂务管理时间

计划过程：

1.列出在开发周期中将生产出的所有产品的规模

根据 START 表和其他 size 数据填写 SUMS 表

2.制定 TASK 计划

1) 计划每个任务所需要的时间
2) 对 task 进行粗略的排序
3) 为每个 task 制定 PV
4) 将这些数据填入 TASK 表

3.制定 SCHEDULE 计划

1) 每个工程师为每个工程计划时间
2) 按周计算团队总计划时间
3) 每个 task 的预期完成周数
4) 每周的 PV
5) 计划经理制定团队 SCHEDULE 表

4.对比 TASK 表和 SCHEDULE 表

1) 团队成员计划的总时间 Statol
2) 将 Statol 与 Ttatol 进行对比
3) 如果 Ttatol>Statol，这个工作对于可用时间来说过大了
4) 如果想引入更多的时间，将新的时间加入 SCHEDULE 表，回到步骤 1
5) 如果决定减少任务，回到策略步骤减少本周期实现的功能

5.完成 SUMP 表的 size 和 time 部分

1) 第一部分 size 数据从 SUMS 表中来
2) 第二部分 time 数据从 TASK 表中来
3) 第三部分 defect 数据从 SUMQ 表中来

6.制定 Quality 计划

1) Summary rates：
   LOC/hour（越高，生产率越高）
   %reuse（复用百分比，用别人的）
   %new reuse（新复用百分比，给别人用的）
2) 无缺陷百分比（PDF-Percent defect-free）：高质量的产品应该在开发进程中有稳步增加的 PDF，并且应该在系统测试时达到或超过 90%.
3) Defects/page：测量在 RE 和 HLD 文档中修正的平均错误数量
4) Defects/KLOC：测量出/入某个阶段的质量
5) Defects Ratios：
   代码复查/编译（DR>2）
   设计复查/单元测试（DR>2）
6) Development time ratios：
   需求检查/需求>=25%
   HLD 检查/HLD>=50%
   DLD/code>=100%
   DLD 复查/DLD>=50%
   代码复查/code>=50%。
7) A/FR（质检过失比）：（review+inspection）/（compile+test）
   在 PSPi 中应该 A/FR > 2
   在 TSPi 中应该 A/FR = 1。
8) Review and inspection rates：
   需求复查和检查：每小时小于 2 页
   HLD 复查和检查：每小时小于 5 页
   DLD 复查和检查：每小时小于 100 行伪代码
   源代码检查和复查：每小时小于 200 LOC
9) Defects-injection Rates（Defects/hour）
10) Defects-removal Rates（Defects/hour）

11) 阶段效益（Phase Yields）：在一个阶段中移除错误的百分比

Eg.如果在进入代码复查有 19 个错误，代码复查阶段引入了 1 个错误，发现了 15 个错误，那么

代码复查阶段效益 = 15/（19+1） = 75%

12) 过程效益（Process Yields）：在给定的阶段之前移除错误的百分比

Eg.如果在编译之前引入了 30 个错误，移除了 20 个错误，则

编译前的过程效益 = 20/30 = 66.7%

在第一次编译之前要努力让过程效益达到 75%,在第一次测试之前达到 85%。

7.根据 SUMQ 表和 TASK 表填写 SUMP 的空白部分

8.制定工程师个人计划

9.平衡团队工作

10.出口准则

第六章 定义需求

SRS（软件需求规格说明书）：

1.Functional requirement	功能需求
2.External interface requirement	外部接口需求
3.Design constraints	设计限制
4.Attributes	属性分析
5.Other requirements	其他需求

为什么 SRS 很重要？

1.用户通常直到使用了最终产品才知道他们真正的需求，所以需求经常改变，只有将需求冻结在文档中他们才不会一直改变

2.SRS 帮助管理变更

需求进程：

1.入口准则

2.需求语句复查

3.需求语句澄清

4.需求任务分配：开发经理带领团队制定 SRS，队长具体分配任务

5.需求归档：每个人制定并复查 SRS 文档中各自的部分，并提交给开发经理，开发经理整合为 SRS 草稿

6.系统测试计划：测试功能和性能是否满足用户的需求

7.需求和系统测试计划检查（质量/过程经理）

a) 检查 SRS 草稿和系统测试计划
b) 识别问题
c) 定义何时何人解决问题
d) 将检查结果填写在 INS 表中

8.需求更新

9.用户 SRS 复查

10.需求基线

11.出口准则

第七章 团队设计

设计原则

1.HLD 必须制定 SDS（软件设计规格说明书），里面包括功能的组件，接口和行为

2.DLD 定义了逻辑结构

3.HLD 和 DLD 仅仅在范围和细节方面不同

4.HLD 在设计阶段执行，DLD 在实现阶段执行

设计标准（NISDLD）

1.Naming conventions

2.Interface formats

3.System and error messages

4.Defect standards

5.LOC counting

6.Design representation standards

设计复用

1.复用接口设计

2.复用文档标准

3.复用部分的质量

4.应用支持

SDS（DAPD）

1.Decide on the overall product structure 产品总体架构设计

2.Allocate product functions to components 产品功能分配到部件

3.Produce the component external specification 每个部件的外部说明

4.Decide which components and functions to develop in each development cycle 确定每个开发周期开发那些功能和部件

设计进程：

1.入口准则

2.高层设计：开发经理带领团队设计

3.设计标准：质量/进程经理

4.设计任务分配：同需求人物分配

5.设计规格说明

6.集成测试计划：只测试各个模块之间的接口和调用关系是否正确

7.设计和集成测试计划检查

8.设计和集成测试计划更新

9.设计基线

10.出口准则

第八章 产品实现

实现标准（SNCSDD）

1.Standards review

2.Naming,interface and message standards

3.Coding standards

4.Size standards

5.Defects standards

6.Defects prevention

实现标准与设计标准的不同：3、4、6。

实现策略：

1.Review 复查

2.Reuse 复用

3.Test 测试

实现进程

1.入口准则

2.实现计划：开发经理把任务分好

3.任务分配：队长将人物分配到具体的人身上

4.详细设计和设计复查：LOGD 和 LOGT 贯穿整个 TSPi，是所有数据的来源

5.单元测试计划

6.详细设计检查：只要有检查就要把结果填写到 INS 表中

7.编码，编码复查和编译

8.编码检查

9.单元测试（UT）：进行 UT，完成 LOGT 和 LOGD。

10.组件质量复查

11.组件发布

12.出口准则

第九章 集成测试和系统测试

测试准则

1.在 TSPi 中，测试是为了评估而不是修正（在测试前找出并修正几乎所有错误）

2.产品的质量是在开发阶段决定的（将低质量的产品放入测试，测试结果也会是低质量的）

TSPi 测试策略

1.构建系统

2.集成测试

3.系统测试

4.回归测试

构建和集成策略

1.大爆炸策略：把所有功能放在一起进行测试

a) 优点：使用最少的测试开发

b) 缺点：很少成功

2.One-at-a-time 策略

a) 优点：能测出新添加的功能是都有问题

b) 缺点：要求大量的测试开发工作，支撑材料多

3.聚类策略

一次添加一类模块，介于第一种和第二种策略之间

4.平面系统策略

a) 优点：能早发现系统范围关联关系问题

b) 缺点：需要有大量相应的子模块空返回为尚未实现的功能做支持

系统测试策略

1.功能第一策略（推荐）

a) 功能测试

b) 在正常条件、非正常条件和重点条件下进行可用性评估

c) 进行性能评估

2.功能区策略：一些相关功能一起进行测试

3.前两种结合策略：先测一个功能，再测试功能区（自底向上）

4.与第三种相反策略：自顶向下

5.

测试计划

1.测试步骤的列表

2.每个测试的支撑材料

3.测试应该得出的结果

4.无缺陷运行时间估计，缺陷查找和每个测试的总时间

5.测试计划中开发每个 item 所需工作估计

跟踪和评估测试

系统测试日志（LOGTEST FORM）包括测试运行统计和包含的结果

文档（DDDARU）

1.Document Organization 文档组织
2.Document Terminology 文档术语
3.Document content 文档内容
4.Accuracy 准确性
5.Readability 可读性
6.Understandability 可懂性
7.书写风格
   a) 使用短句
   b) 使用简单的词语和短语
   c) 使用大量列表和专用条款（bulleted items)

测试进程

1.入口准则
2.测试开发
3.构建
4.集成测试
5.系统测试
6.文档
7.出口准则

第十章 事后剖析（postmortem）

为什么我们需要事后剖析？

1.事后剖析帮助个人和团队进程提供一个结构化的方式
2.TSPi 用 PIP（进程提升提议）表标识你想出的改进想法

事后剖析进程

1.入口准则
2.复查进程数据
3.估计角色性能：是否高效，是否有提升空间
4.准备第一周期报告：内容，统计，角色报告（五种角色），工程师报告
5.角色评估
6.出口准则

第十一章 Being on a team

管理自己

1.负责人

2.努力达到定义的目标

3.有原则的生活：尊重自己，尊重他人

团队工作义务

1.和其他团队成员交流

2.做出并达到承诺

3.参与团队活动

团队建立义务

1.接受团队角色的责任，并尽己所能

2.参与建立团队目标和计划并尽力达到

3.构建并维护一个高效合作的团队