1

テストプロセス

オフショア
もうまく
使えない.

SP
~~イテレーション~~期間内
に終了しないで
雪だるまになる
課題が
のでは

テストフェーズは
どこに組み込まれる?

品質の判断は
どうやってする?

テストケースの
もうら性が
わからない

品質/成果物の
見える化の弱
進捗の
見える化

プロジェクトの価値.
制約
へ
顧客

品質レベルは

仕様変更に対する
テストの品質保持
はどのように保つか?

何を元に
テストするの?

最終的な
品質の確認は
どうする?
~~何を元にテスト終了?~~

開発チームの
テスト能力.
レベルは
(リスクは?)

テスト技法
手法

個人

組織

2

2.
イテレ毎のテストで、
単体テスト・結合テスト
の強化。

1.
スプリントバックログ
のタスクを見積り。
ベロシティーに沿った
イテレーションの量に調整
変更な

スプリントのテスト
品質を確認する

イテレーションの中で
QAが品質をFeedBack.
する.

・各スプリントの
バックログの洗化
(直接)とベロシティ
を見える化する

・全バックログの終了
又はリリースまでに入る機能

仕様が比較的
堅い機能から
先行してシステムテスト
を前倒しする.

システムテストで
やる事を減らす…

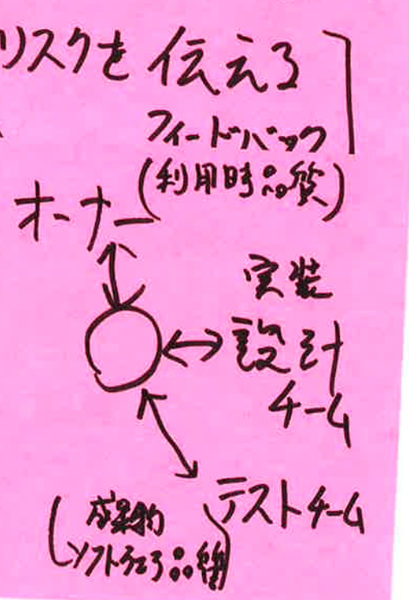
3

・仕様書、や
単体テスト、結合テスト
を作って実施して
欠陥を

情報をだす
⇄情報を取りに行く

何を得たいのか
明文化して提示する

プロダクトバックログ
をベースにコミュニ
ケーションする。
[リスクを伝える
フィードバック
(利用時品質)]



1

納期 コスト

イテレーションの期間に実装が完了しない場合は次のイテレーションに持ち越しになると、雪だるま式に後工程が苦しくなるのではで？

バグ発見時のスケジュールの対応はどうするか。

期間は、短かくない？

テストタイミング

人員配置

変更していない部分のテストはやるのか？

1人1人

具体的に何から？

不安

テストで完璧？

変更点/追加の仕様はどうやって共有するか。

評価方法は？

CI化作成する時間ある？

探索型テストにする？

2

イテレーション中に
開発と並行して
スプリント中での
作業量を調整する

評価を早い段階から
実施.
→ 仕様が固まらない
内にテスト設計出来る
か?
→ 設計から参画?

実装した部分から
事前にテストをし
FB.

QAチームのによる.
仕様書の作成.

イテレーション
中にテスト分
析含めて.
テスト完了迄

増員

3

開発にQA(テスター)が入る
メリットを説明する
(入ると負荷になるという
思い込みを消す)

バックログから
情報へたどる方法の共有.

・設計を手伝って
もらえるか
文書になおして
あげる.
・指定して
テストしやすいように

システムテストより前の
結合テストや単体テスト
を実施してあげる
その代わり負荷が軽く
なった開発にドキュメント
を作成
仕様に関する
してもらう.

品質を
どのように
保証するの？
レ

サイクルごとの判定
はどのように
するべき？

サイクルごとに
達成すべき内容
は何？

テスト

・テスト設計、計画

イテレーション毎
にコーディングが
遅れる
レ

そもそも
何が課題？

アジャイル
導入すると
良くなるの？
なぜか、、

個人

ここでの
Agile って
何？
レ

要求仕様の変更に
どう追い付くのか？

お金かかる？

人かかるの？

新しく覚えること
が多くなる

よく
テグレ
するね

構成管理の
手法は
どのようにする
の？

導入
手法

2

ベロシティを断念し
無理なバックログに
なっていないか確認する.

出来ないモノは
出来ないと言って
仕様を考え直させる
再度

仕様の追加.
予算を.
あきらめる
(出来ないものは.出来ない)

ボトルネック的な
チームに対して.
何が問題で.何が
原因か.一緒に検討
する.

fixした仕様を
事前に探針
テストをして
リスクの早期明確化

おうだん的な部分の
テストは自動化して.
常時一定の品質を
保持する.

状況の変化に合わせた
システムテストの計画
みなおし

開発者を
せかす.

過去のバグを
分析する.

人員を投入
する
より良く
(コスト無視)

3

コミュニケーションを
把握する。
(人間関係とか)

仲良くなる
(飲み会とか)

飲みに
行く

仕事以外のことで
仲良くなってみる.

設計者の負担を
増やす事を言わない
(改善は対価を見せ
提案ベースで)

信頼土台に必ず
何かを持つ
{スキル
経験
など

設計書みたいな
ものを自分で作って
レビューをしてもらう.

日頃から.
1対1の場
(飲み) (ランチ)
付きあいをする
(感情が大事)

現場の課題を
持ち、率先して
解決する事で
信頼をえる.

とりあえず
相手の話を
聞く.
自分のことばかりじゃなく.

テスト

2週間の内のテスト日程

(品質保証) リリースするための基準は？

テスト対象のリリーススケジュールは守られるのか

プロセスをメンバにどう浸透させるか？

2週間の中で、いつからテストを開始するのか？

個人

組織

ウォーターフローのテストとの違いは？

QAとして、どう開発スピードを落とさないように関わるのか？

テスト結果がたくさんNGになる原因調査が大変。

どの程度作成された段階でテストするか(始める)

テスト内容は？

テスト

テスト

テストの順番をどう決める？

2

バグを作り込まないようなアクション？
Prjの初期から関わる。

バックログを理解して早めに突っ込んでおく
(初期のイテレーションから)

あらかじめ、テストの内容を共有して、実装時に意識させる？

テスト設計を早める

要求に対してどの程度のテストの総量になるかを予測・提示。

イテレーション間のゲートを厳密にする。

開発者のテストをレビューしテスト不足を防ぐ

テストに影響が大きい機能をあらかじめ共有

Unit テストを実施してもらって、単体レベルのバグをつぶしておく

静的解析ツールを導入して、しょうもないバグはつぶしておく

3

心理作戦：
開発者は教えるのが好きなはずなので、「教えて下さいよー♪」としたてに出ながら情報をキャッチ.

人として信頼を得る.

開発計画時に合意を得る.

トータルで皆が幸せになれるよ.ということを提示する

ソースコード読めるようになる

1

テストプロセス

③
要員計画
が難しい。
(入りと出)

人の役割。
(責任分担もない)

テストで、
どういう状態
ならOK？
何は後回し？

テストのスケジュール
(テスト可能時期)

製品
開発
スケジュール

② 何から ~~開発~~ (実装)
テストするのか
優先順は
どう決めるの？

要求仕様が ㊀
あいまいなん
じゃない？

個人 ←

エビデンスを残す
テストのりれき管理

テストの
回数。

自動化を導入するか。

・改修 終わらないのでは

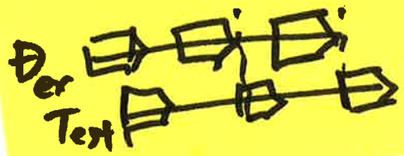
・仕様変更 止まらないのでは
・~~改修、まとまらないのでは~~

テスト技法
手法。

2

・要求をチェック（テスト）

イテレーション毎に部分的にテストする



・開発の横ですぐテストする

人を増やす。

成果物を減らす。

やれる範囲をしぼる

開発中の仕変、追加要求が発生したら、必ず再見積やスケジュールを見直すこと

バグ（異常となるパターン）をキョウセイする。

（バグいじり設定？）

3

権限のある人から必要性を説明する。
チーム間に(仲良くなりそう。)

・「プロジェクト成功のためだから」
と説得

対象：製品
・Dev → 作る人
・Tester → 見る人.
(伝える)
↓
~~を分ける.~~
見・伝え続ける

単体や結合のテスト仕様を考えたり提供してあげる
テスト駆動型で思わせる.

テストが終わらないとスケジュール通りにリリースで強行

仲良くなる。

口出しせず
助言

テスト設計にかける時間はどれだけとれるのか?

テスト設計がやりずらい.

テストプロセス

テストケース作成のためのインプット不足

機能のデリバリ
がIterativeだ
システム全体の
テストが一括で
スタートできない

会社の指標(WFの指標)だけなので、どう整理するか

品質保証の定義をどうするのか
(仕様が明確でない。ここをどうする)

個人

組織

本番で不具合がでた時に2ーずん説明ができるか.

評価項目をどう定義するのか
完了条件.

その人が休んだ時にテストができない. 進まない. 遅うものにするのではないか?

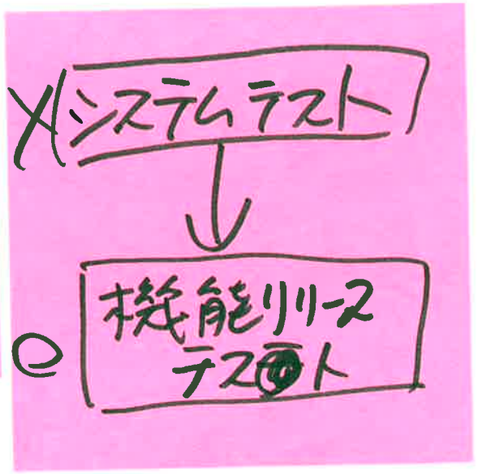
自動テスト

リグレッションテストが増える.

テスト技法
テスト手法

2

デリバリー単位のテスト実施に切り替える



各イテレーションで
溜まったバックログを
なくする.
リリースから落とすかも.

イテレーションを
短くして、テストの
時間を増やす

イテレーションの
数を減らす.

イテレーション中に
あるレビューを
省略する

① 開発が終わらない理由を明確にする
↓
対策ができれば
次スプリントでやるか
POに相談

② スプリント中で
開発が終わらない
場合.
次のスプリントで
とりをやめるかを
明確にする.

3

ビールを飲みに
いく.(メンバー)

共通で使う
モニターを
準備する
(大きいサイズ)

席をとなりに
移る.

ひたすら
傾聴.

コーザストリーをみずから
かくに入し、自律的にうごけるように
努力する

ネタだり下さい.
ドキュメント化はこちら
でやります.

更者は最小限にする.

1

どのレベルまでできればよいと言ってよいのか.

何をテストすればよいのか明確になるのか

開発 → 評価 → リリース の工数配分

変更管理をどうする?

・テストの実効化

・ゆるい設計に対する品質

仕様・設計がきちんとDocument化される…?

不具合の管理

アジャイルに合う品質保証とは

・スプリント毎に実施する?

・品質保証をせずにリリースをする

・どこまでを対象とするか.

個人

組織

テストケースの件数はどう考えて設計すればよいのか

テストリソースは充分に実施されるのか.

・単体テスト
・結合テスト
どのようにまわすか

テスト終了の時間はとれるのか

2

イテレーションごとに
テスト行
と一連でテスト

1つのタスク完成ごとに
テスト依頼書を書いてもらい
それにもとづいてテストする

不具合原因の
分析と対策

1イテレーション終了時で
積み残しタスクが
あったら、リスクを
検討する。

終了基準（計画時）
にセキュリティを、設定
して開発を進める

開発側テストプロセス
管理を
サポート

QAとして。

開発前に
要求仕様を
固める

要求仕様～
不具合作込み
減らす。
→要求仕様のレビュー

3

QA視点の
設計レビュー

となりで
お仕事

お互いの
実態をさらす.
→お互いの課題解決
の手助けになれば
という考え方

せいけいの資料を
よんでみる

次にどんな仕様を
テストすればよいかを
かくにんしつづける

テストの論点と
スプリントで
まとめてもらう

・Dev目線
・リリースの自動化
・Test Automation

「仕様書の行間を
私が埋める」
という気概を持つ

I

設計が決まっていないので各人バラバラで作業

設計者との関係，テスト移行（解釈）

テスト
・SPRINT内
・システムテストはウォーターフォール/アジャイル SCRUM型？

仕様変化が多発してテストがおいつかない

仕様書って？

仕様変更のサイクルが早いor テスト設計の時間がとれない

工程がどの部分か把握しにくい

従来と違うやり方で何を基準に品質を測ればよいかとノ今ぐた…

ドキュメントがないのでテストするのが大変

テスト件数はどれくらい？

テストケースの管理 Revision

全体像が見えない 終わりが見えなくてひろう

レビュー/設計のメトリクスがとれない
→妥当性がわからない

テスターの確保

1週間あたりのテスト工数はどれくらい見積るのか？

パターン抽出をどこまで実施する？

自動化のコストが大きすぎて途中で頓挫

影響度（ソース範囲）を開発者→テスタにどう伝えるのか？

最終的に
何をつくるべきか定義してない…ので内容，品質管理が困難
テスト（やり直しとか）

2

イテレーション内
テストの対象
を特定する

バグ改修の
ための
情報開示
(不具合の状況)

探索的テストで
おかしそうな所を
作りながら
あぶり出していく

W字で仕様の妥当
性検証など
リスクを前出し

イテレーション
SPRINTの中
でテストし、
その度を都度に見直める.

スプリントバック
ログにつめこま
ない

最終的に
達成してほしい品質を
提示しておく

システムテスト
の早期設計

イテレーションの段階から
システムテストに近い
レベルのテストで
品質を上げておく

3

次に着眼点での振り返り

設計の負担を下げる
(デバッグを行う)
計測りを行う

設計の契約で困り事を解決する。

朝会に参加する
着眼点を増やす
(今からでもいいから)

・一緒にやっていこうというマインド
・開発を楽にいいものをつくっていこう!

1

個人でやることが不明.
自分勝手でも?

個人

開発との連携をどうやるか?
(どこまで情報を共有するか)

ドキュメント類が心配.
何を基準にテストや分析は?

第3者検証を維持できるか?

テストの終了条件をどうするか(範囲)

イテレーションで障害修正はどうする?

テスト技法
テスト手法

仕様が決定してから、テスト開始までどれぐらいか?

テストサイクルはどうまわすのか?

評価スケジュールがたてられない

カスタマーにどう納得してもらうか?
(品質保証)

QAの入るタイミングとその役割は?

全機能評価の実施時期がわからない

要件定義・仕様のとり決めはどうやるのか?

評価項目がわかりにくい

テスト項目の洗い出しはいつやるのか?

2

イテレーション毎にテストを行う.
※テストは少しずつ増える形

イテレーション毎にシステムテストを行う.
実装済の品質保証

イベント毎に基本機能から評価し、不具合をやばそうな方から見積もれるようにする

開発スプリントにQAの数を送り込む（一部）

仕様の打合せに参加して、早くから理解する.

プロダクトバックログより.懸念点を挙げてフィードバック

初期にバックログのインフォメーションを得て早くにテスト分析を行なう →スプリントの早期に提示.

機能、動作が明確になっているか確認する.

システムテスト時の計画を示すことで、リスクを共有する.

開発計画を立て、イベント毎にリリース項目を明確にして、遅れを早めにわかるようにする

仕様書の提出をせっつく.

テストをする人が開発もできる様にする

3

御用聞きをする
⇒話をしやすい環境作り。

ドメイン知識を身に付ける
⇒話題についていく

暗黙知＝認識ズレを防ぐために。
分かる用語資料を作る。

最終的にはドキュメントが確定する事を合意したうえで、プロセス中は受け入れる。

信頼を得る迄は意見を言うのではなく、聞く事に専念する。

開発の設計を理解できるようにスキルアップが必要

相互の歩みよりを目指す。

設計の情報を盗む
(手間をかけさせない)

1

責任の存り処.

~~わりあて~~
作業順番

ドキュメントのフォーマット
などはあるのか

何が変わるの?

スケジュール

プラクティス

メンバ
役割ソ

教育は
どうするのか.

スケジュールはどうな
るのか

わりあて

個人

仕様(書)

どうやって設計してい
くのか

手法

2

バックログが粗い段階で
設計する.

イテレーションに
テストを入る。

開発に入って
イテレーションの
中でテスト

早い段階で..
テスト開始
.

終わらない機能は次回
にもすこし

システムテストの
開始条件の見直し
は出来ないか?

旧回会議
で問題点を
あげる

3

どうしてそう言うのか説明して
その仕事はすべてQAが行う
- 客観的に伝える.

テストエンジニア・QAが設計書を作る

最低限のメトリクス
↓
キックオフで説明してみる

(テストレビュー)
コミュニケーションをとる

ヘッダファイルなど
出してもらえそうな
ものから出してもらう.

情報の
共有を

1

テストプ

仕様書は、どこまで、作成するのが？

テストするに足りうるものが提供されるのか？

充分なテストの時間がとれる？

アジャイルは良く聞くが、良くわからない

仕様書はあるのか？（ドキュメント）

2週間後のゴールがわからない

個人

アジャイル開発の経験がない

・仕様確認を、いつ誰と行なえば良いか？

組織

・客先データの収集も自分達の責務になる？

・~~ユニットテスト~~ モジュールデリバリーのテストは楽になるだろうが、それ以外の部分は？

テスト技法
テスト手法

2

積み残しが起きる
原因は？

作業の範囲は
期待しているものか？
想定

言什定義を
はっきりとさせる..
優先順位も等.
おいておき.
すりのでいように.
サポートする。

無謀なスケジュールを
組ませない。
(プロジェクトキックオフ時に)

テストする前に
見積りする

・システムテストの分割,
イテレーション内のテスト

3

システムテストの必要性
を理解してもらう
伝える.
BDS

暗黙知を
共有するための
開発に対する知識
(テスト対象の構造に
対する知識?)

内容の対して
何ができるかを考える
(QAとして).

テスター など第三者が
入るメリットを説明して
納得してもらう。
(バグを早く発見できる etc)

・役割を
明確にする.
テスター → アウトプットを出す.

1

メンバの
教育はどうするのか？

プロセスについては
1から検討していく
必要もあるが、
細かく確認していくので品質は担保できそう。

短期向け：
効率の良い
テスト体系

環境
テスト手法
手法

個人

ラストプロセス

仕様変更へのカバレッジ
対応　が取れるか

~~カバレッジ~~
バグの修正
タイミング

どのタイミングでテストするのか

リリースなどのタイミング
（イテレーション目）を目指すか

テストの範囲
回帰のレベル

テスト環境の準備
& 自動化レベル

2週間毎に
何がリリースされ
どこまでテストすれば
よいのか？

デグレード

リグレッションテスト
はどう進めるか

2

残タスクの
開発との相互
検証と
自動化項目の
決定

バックログが溜まれるのは
~~スクラム~~が悪いのではなく、
前工程から、ポイントを抑えた
作りこみができるよう
開発側とQA側両方の
観点で作っていく

スクラムマスター、
開発、QA、いずれも
スキルUPする

実施できる
MAXのテスト項目を
きめ優先度つけて、
できる分を出荷。

1~~イテレート~~の
イテレートが終了した
時点でテスト設計。

開発側との
密なコミュニケーション
フィードバック

新規
柔軟で
~~全~~必要のある
テスト計画
スケジュール

静的解析エラーが
出てないか監視し、
日々是正してもらう。

顧客とのMTGで
議事をとり、合意を
とっているかチェックする。

3

一緒に
物送りをするという目的を
共有する。
ゴールの共有。

テスト・QAとしての
設計観点を入れることで、
後工程の手戻りが
出ないことを
理解してもらう
(スクラムマスター含め)

イテレーションの
成果物として
定義する

・どこまでやらテストしたら行き成果か
・○○という仕様は
といったような、つめる話し合い以外で
開発側に持つ

QAからも
リリース前のテスト協力
などで協力体制を
築く。

1

役割がどうなるか…

テストプロセス

イテレーションが多くなるので工数増大！

デグレが増えそう

成果物の管理方法。

ソフトリリース後の環境含めた準備工数は？

スケジュールに遅れた場合

チームの役割

個人

どんなスケジュールで進めていくのか

テスト
・期間が短い
・~~長い目で見る必要が~~

テストケースはどの粒度の段階で作成するのか？

・長い目で見る。

テスト側の工数／リソース配分は？

・どんなセアリング(?)で理解・評価していくのか
（技術の確立）
・~~イテレーションの期間でテックする間に合うのか~~
~~継続するそう…~~
・~~どんどんプロセスが進んでいった時、以前の結合をデグレが増えたりしないか~~

テスト技法
テスト手法

2

上三流で不具合を指摘する

C上流工程へ関わる.
スタートから

QAもシステムテスト前からテストに関わる
必ずレビュー毎に情報共有

日々の進捗管理

作業の見積りを正確に.

無理なスケジュールを立てない

予めバッファ期間を設けておく.

リソースを増やす

バグの重み付けを変える
(直す/直さない)

3

~~この分の負荷を~~
負荷が軽くなるよう
サポートする
一部でも
ドキュメントを書くとか

仕様を増やす提言
だけでなく、
削減も

理解してもらう
せっとく。

次めて 説明会。
打合わせ
を開く

QAが入っていく事での
メリットを開発側へ
しっかり伝える(説得)

開発側の
意見もしっかり聞く。

情報/リスクの共有化
開発側とのコミュニケーション
を図る

少しずつでも
メリットを実感させる

1

そもそもアジャイルとは

テストプロセス

2weekで
テストが終わるのか？

目的は？何を目指している？

テストできるモノはいつでつくる？
（本当に2週間後？）

発見した不具合は修正される？放置？

- イテレーション内のテスト期間は？
- ~~終了~~条件は？
- 終わらなかったらどうするのか？

テスト仕様作成
テスト実施
間に合う？

個人

組織

テスト技法
テスト手法

2

1、見積りん勉
2、仕様検討に参加
3、テストを肩がわり

積み残さない様に仕様を割り切ってFIXしていく

イテレーション毎にテストを実施する

開発と同時にテストを実施システムができたところから

スプリント内でテスト可能なことを開発も意識合わせする.
(スプリント0)

3

システムテストのテストケースを作ってレビューしてもらう。
(きちんと理解しているか)

朝会とはなからなくてもよい。
曖昧な部分は分からないことをこちらから明確にしていく。

・設計レビューに参加する

開発チームを助けたきたと説明する

メリットを伝える。
(工数面など)
否定的にならない。

朝会ミーティングに参加し、QAチームの観点でアドバイス・指摘し、Give and Takeの関係で、信頼関係を築いていく。

1

プロセス

既存の
テストチームを
どう分けるのか？

どこまで
テストすれば
よいのか？
（開発との
コミュニケーション）

イテレーションに
遅れるときの
調整

設計した
テスト設計の
回数

品質確保
どうする

品質保証の
責任の所在

品質保証の
基準

個人

組織

アジャイルってなに？

今までの
成果物を
どう使うか？

詳細な仕様書
がない or ないから
テストケースを考える
のが大変

ソフトの仕様が
ひんぱんに変る
からテストプログラム
のメンテが大変

既存の
テストケースを
どう利用するか

ひんぱんに変る
プログラムの品質
保証をどうするか

技法
手法

2

イテレーション中に
テスト実施
→バグ出し
でれば良いが…

システムテストに入る前の
イテレーションの段階で
並行してテスト
できないか？

システムテストのみを
何回か試して
精度を高めてゆく。

実装されてからの
テストではなく、
要求仕様だけが妥当か
チェックする。

3

少いdoc
ソースコードから
テストプログラム.
を作成する.

これまで
やってなくても.
朝会の
議事録を書いて
配る.
(・指摘が来たり
・用語が統一できたり

1

テストプロセス

終わりはでこ

ドキュメント
(仕様・設計書)が
なくなる?
⇒テストベース
どうする?

いつからテストをするのか
開発のどの段階から関わるのか
次のサイクルの時に前の
テストはどうなるのか

テストのインプット
は確保されて
いるのか?

個人

バグ修正時の
影響範囲分析が
難しくなりそう

2週間のうち
システムテストは
どれくらい期間
があるのか?

十分な期間
が
確保できるか

テスト→観点はどうする.
サイクルがまわったら.

テストエンジニアに
求められるスキルが
上がる?

スキルが足りて
いるのか?

アジャイルって何?

テスト技法
テスト手法

システムテストの
自動化を
進めないと…。

2

過去の不具合
事例を整理して
開発者へフィードバック

テスト仕様を
先行して作り
開発と共有する

どういうテストを
したいかをインプット

テストケースと先に
開発に提示する
(TDD的?)

開発の早い時期に
テストを開始
(仕様のレビューでも)

スプリント毎に
システムテストを
実施する。

テスト工程の前倒し.
初回の作しーダから
テストする.

レビューの内容
を確認して
テストの優先順位
を決めてみる.

ウォークスルーで通し
のテストをする.
(最終作しーダ直後に
バグ検出しや)

UIレビュー参加
(UI変わったら悲惨)

3

あげ足は
とらない
リスペクトする.

まずは
開発者の話に
耳を傾ける.

設計者の会話
にかたよった傾向
があれば、そこに
突っこみを入れる
(内省をうながす質問)

過去のバグ情報が
今回混入するであろう
過バグと提示
する.

設計チームが
考慮していなかった
ケースまで想定して
バグ混入を防ぐ

トータルのコストの話をする
(品質を上げておけば、
初期のコストはかかっても
ペイする)

テスト側の
観点を提示する

一緒に
仕様を作る

酒

飲み会に行ってみる.

品管の妥当性
(比較ができない)

・メトリクスは
どうなるの？

・体制(人数)
・各

テストボリューム

キーパーソン
になる
彼に任せれば
安心できる
キーパーソン

・テスターはアジャイル
のイテレーションに参加
するのは難しそう.

・テストがじゃんぱんに
なり大変だ
そう

体制.
(自院テスト
の運用)

相手のメリットを考える.
伝える
⇒

ツール
(コミュニケーションツール)

・テスト設計と実装
の期間や
必要性

必要なコンピテンス
比較がしずらい
(品質の妥当性)
体制)(開発規模
のぎり)
ツール

テスター スキル.

テストに
必要な
コンピテンス

必要な
・ドキュメントは何が
ある

バグ報告方法

2

・次イテレーションへキャリオーバー無し
・スペックを切る
・イテレーション延ばす
・

要件，リスク，影響範囲を備前に不安分子洗いだし inputする

・イテレーションごとの要件の部分はイテレーションごとにテストする

品質レベル
・定期フィードバック
・レポート

・1年間は修業と
思って入る。
暗黙知の取得。

彼らと同等レベルの
コンピテンスを持つ。
組織のキーパーソン
になる。

オーナーからの要望はたくさん.
スプリントは2week.
開発の人手不足で、システムテスト以前にテストは出来ない.

プロダクトのバックログ(~~テスト~~)はあるが仕様書なし.
テスト設計どうすんの?

テストベースは何?
ドキュメント作るのか

組織にマッチするか不安

1日の使い方
・テストのタイミング など

個人

進ちょく管理は?

顧客との交渉(かかわり方)

組織

仕様をどうまとめていくのか?
(なし崩しに決まる?)

何度も同じテストするの?

回帰テストはどうする?

さいごに全てテストする?

テスト自動化の可否

アジャイルって何?

@(テストを考えると)
2weekって長い,短い?

テ

改善するために

ベロシティの測定

システムテストをやめ、開発中に仕組みとしてテストする

安定コードで日常的にテストを自動で回す。

スクラムをきちんと実施する（毎日1回）

進ちょく把握

管理職支えて人の投入を考える

仕様や不具合物を定期的にレビューする

ムリする時のバックログ調整（どこを削るか）

（品質をテストで確認するのでなく）テスト前に作り込む。

ユースケースを作って開発・テスタでレビューする

テストの前にペルソナをちゃんと決める

要求分析をレビューする。

どうチームに入るか

プログラムをやる…?
(テスターが…)

プログラマとトレード

開発チームより他
オーナーや
アーキテクトと
先に仲良くなる。

~~チロアが~~
近くに
席を作る／
用意してもらう。

文書でなく
口頭での
コミュニケーション

そもそも
Dev と QA
分けるべき？

仲良く
なる

共通の敵
を作る
顧客、マネージャ
エンドユーザ？

クレームや
無理な要求を
フィルタ番して
あげる

「お前
設計？と
だよな」
言われるくらいに
なじむまで
しつこく入りこむ

テストプロセス

テストの戻りを
期間内に
どうおさめるのか

検出した問題
の修正、確認
はいつやる？

リースに間に合うの？

アウトプットは
どうなる？

テストいつ
行うか？

スケジュール
の
たて方、
みつもり方、

テストに何が
インプットは、どうなる？
ードキュメントはあるか
ー入力提供タイミング

個人

ユーザーバックログの
優先度を正しく判断
できるの？

どのようなレベル
の仕様書が
あるか？

アジャイルでやるような
粒度に分けられるの？

システムテストは
いつから開始
できる？
（開始タイミング）

どうように
テストするか

手ジ

2

ボトルネックじゃないなら
無視する

こまめに
テスト~~ダブル~~を
システム
はさむ.

システムテスト
の
振舞すr テスト
を毎日

バックログの適正
化→積み過ぎ
ないように
積む時にいっしょに
参画する

イテレーション
最初に
テスト仕様
明確にする

イテレーション
でのテスト(?)
内容をcheck
する

velocityを計る
手法を考える
↓
スプリントをのばす

スプリントで
オーバーコミット
させない.
管理役をつくる.

あふれた時に
都度,
優先度み直す.

3

設計レビューに参加する
Documentは要求しない
口頭で理解する

入る目的今(良い品質のソフトを作るため)を明確にして、不安をとりのぞく

システムテストで
NGでても
「あーそれまだだから」
を受け入れる

テスターはスクラムの
メンバー(開発者)から
だす

一緒に
悩む？

作業を増やさず、
過去のバグ事例
を共有する.
(似ている設計の
機能とか.)

入手した情報を
先行してテスト
する

1

テ

バグの
収束は
どう見るのか?

いつ、誰に、
どな目的で
デリバリーなのか
が明確か?

個人

組織

自部署の本来
業務の[redacted]に
干渉される

短期間で
サイクルを
まわせるの?

環境は
同じか?

開発チームや
アジャイル開発
の経験が
あるか?

テスト技法
テスト手法

開発の人数は
小数か?

2

早い段階で
開発と連携して
進められるところから
進める。

リスク
ベースド
テスト
(洗い出し)

要求の
順位づけ

上流レビュー
参加。

完了の定義
を合意して
守る。

オーバーコミット
メントによるプロセス
の破たんが
起こないかを注意する。

プロダクトオーナー
を支援して品質の
情報をインプットし
バックログに反映する

3

POと役割について合意する

・品質を高める
・品質を確定させる
この2つの目的を明確にする

テストはOKな所
まだできない所を教えて!

アメを持っていく

・何をテストする？
・優先度は？

いつから
テストする？
開始

テスト
開発開始
時期

テスト
プロセス

スプリントの
テストが…
完了するが？

ストーリーや
バックログの
テスタビリティー

テストの
進捗報告の
方法は？

個人

アジャイルとは
を
まずは調査
しなくては
いけない？

・テストに必要な
スキルは？

プロセス
従来のやり方を
全て変更しなくて
はいけない。

バグ対策は
次のイテレーション
のバックログに
つむの？

レグレッションテスト
はやる？

テスト手法

要求の
ゆれ
在庫不足

開発成果物
テストベースは
何か？

2

プロダクトバックログの質を上げる

ストーリーの完了条件

見積り
しっかり

早く
評価に
着手する

ドキュメントからの
バグを
みつけて
早めにつぶす

各イテレーションの
時点で
積極的に
関わる

イテレーションごとに
できているものを
テストして設計へ
フィードバック

3

(QAの人からは
言いにくいでしょうか)
設計チーム、開発チーム、QAチームで
知識を共有する土台を作るべき。

テストで発生している問題をJIRAだけで課題を共有する

コミュニケーションをとる
(ハロイフォをつかう)
えらい人
飲みにいく

バックログの分析、フィードバック

開発計画のキックオフ

スプリント計画から参加

負荷にならないようにQAが参加する。
Win Win

QAが入る事によるメリットを説明する。
品質か、納期とか…

ソースをもとに分析、フィードバック

テストプロセス

ストーリーの
規模は
どれくらいか。

受け入れ条件
明確か？

開発チーム、
テストチームの
ベロシティは？

プロダクトの
全体像がつかめる？
製品の最終版の
イメージ？

どこのフェーズで
テストエンジニアを
参加させる？？
仕様検討段階から？？

プロセス？

品質の計り方？

アジャイル開発
プロジェクトに
おける
終了基準は？

時間（納期）
が足りなくなる。
（テストケースの抽出）

テスト技法
テスト手法

それを聞きたい

開発と同時に
テスト設計等を
始める(QA目線大事!)
できる環境なら…

システムテストで実施する
他システムの連携部門
等の仕様について、
事前に調整する。

イテレーション時(スプリント)に
テスト設計を実施
(システムテスト用)し、
その観点をスプリント内
のテストにとりいれる
(内容入れて)

ベロシティの
実測を行ない、
終了期間を見込む。
東中

イテレーションの
開始時に、
終了基準を設ける。

スプリント・バックログ
内のシナリオが
doneする毎に
テストを行う

設計も手伝います
(実はテスト設計)

テストベース(仕様書)を、要件と実装から書いちゃう。
で、[illegible]。

設計内容を理解する。
要件と実装をリンクさせる。

開発がもっている情報は、
全て共有する。
それ以外何が必要？

・Meritを説明する。
・開発者の上司とコンタクトを。

酒

題1.

2015.2.20.
Tokyo.

テストプロセス

アジャイルって
何かわからないので
教えて下さい.

アジャイルて
何?

実装はに
合う?

バグ修正の
やり残りが
大きくなりすぎて
次スプリントへの
負担が大きくなる
ないか

開発とのコミュニケーション

評価が矢盾くな
る?

不具合報告は
今までと異なるつかい?

しれぇうほ?

バスクライテリア

スプリント単位で
どのレベル(品質
基準)・テストの
終了基準は?

個人

組織

テスター
スタッフ
リソース
確保できる?

非機能テスト
はいつやるのか

テスト対象の
単位・順序
(レベル)

手法

その他

テスト環境は?

ドキュメントがあるのか

ドキュメントの質

題2.

2015.2.20
JaSST.Tokyo.

上流工程での仕様レビュー（不具合の早期対策）

積み残しの共有。どれだけ残ってしまっているかを開発者やマネージャに報告

システムテストをもう少し細かくしてイテレーションの中に入れていく。

イテレーション単位のスモークテストをする。

イテレーション毎の障害分析で品質予測

前だおしで、システムテストをかいしバグをたたく

問題のきりわけをやる

れようかじゅんばんをいれかえる？

QA側による開発側タスク管理．進捗

ストーリへの工数見つもり、テスト（早期警報）

単体システムテスト
統合システムテスト

題3

JaSST Tokyo
2015.2.20.

疑問点は聞く。
話題が暗くでると
感じたら聞く。
設計者の負担に感じ
作業を一部肩代りする

テストベースとしての
ドキュメント整理
役をしていく。
(形式化)

えらい人を使う

仕様書はこうで
作る。
事件と繋しれい
へ機能 事を
教えて欲しい
負担を請け負う

フィードバックさんは
情報と効果
報 を説明
(目的)

作業をてつだう。

酒

1

2週間で
何もつくれない
気がする？

テスト工程をどう扱か？

最終納品時に
特別な対応は必要？

ドキュメンテーション
どうする？

言葉と一緒に
やるの？

開発者がちゃんと
テストしてくれるのか？

品質はどう作るのか？
いつ？
誰が？
どうやって？

・テストに多く工数が
かかるので
開発は進むのか？

個人

組織

・アジャイルがよくわからない人がいる
・~~テスト工程をどこに持っていくのか~~
・~~開発工程の割合が小さい~~

テスト自動化が
前提なのか？

アジャイルで
リグレッションしないか？
どう~~手~~同じテストを
繰り返すのか？

・仕様はどのくらいで
変更
・誰がどういう風に
仕事をもつか

設計はいつ誰が
どのタイミングで決める？

変更影響が
明確ではないのでは？

同じテストは
何度も繰り返さ
なきゃいけないのか

手法

Q、アジャイルやるよって
言われた時のケネン

2

○プロダクトバックログの1個1個のストーリーにDONEの定義をテストとして書く
⇨テストのバックログをもつ

○スプリントバックログにテストのバックログをいれる。
⇨テストをスプリントで実施

○最初にデプロイ

・前にでた不具合情報のまとめ。うくせを開発にわたす
対策をおこなり上にする

・スケジュールの見直しかんりを行い、ヤバそうな気配を感じたら上にアラートを出す

○スプリント中のテストは極力自動化
リグレッションテストとして実行

イテレーションに合わせたシステムテスト設計とする。
↓
イテレーションごとの。

e.g. ODC
バグを分析して、どんな低同の品質かチェックする

開発のIterationのテスト内容もチェック、改善要望をする

リグレッションテストを自動化する

システムテストも一定周期ごと実施する

欠陥を予測・予防する

レビューしよう!

Q. システムテストに入れない、が終わらない を防ぐには？

3

設計にとって、
QAは敵でなく
味方だと
認識させる。

情報を出したことによる
設計者のメリットを
伝える。

負担にならない (アドバイス)
設計者の負担にならない
フィードバックを与える

予想・予防 比例的な
知見、など。

具体的な行動に"すぐに"
つながらない、アドバイス

・QAが開発のミーティングに
参加して、テストケースなどを作る
テスト仕様書と

→ミーティングの書記ぼりにも。

→あいまいな点をはっきりさせる
書類化する
QAが

開発工程のすりあわせ

開発者の視点ではなく
エンドユーザの視点で
課題/問題を指摘

開発し易く
なる情報を
提供し続ける

・信頼
価値ある会話が目線

ニコ
カレ。

Q. どうしたら設計/開発チームに入れるか？

スクラムこんで
いつテストするの？
テスト中の不具合
出たらいつ修正する？

テストのライフ
サイクルは、
WFとどう違う
か？何のテスト
から始めるか？

誰がテストする

どこまでテスト
すべきか？

~~どこまで行く？~~

やらない保証？

体制は、
どう考えていく
か？

メンバーへの周知の
方法は？

- テストの生産性は？（束縛されない）
- どこまでやるか？
- やらない所の保証は？
-

個人

何が仕様か
分かるの？

プロセスが
理解できない
（認識合わない）

テストの
生産性も？

アジャイルと言われて
もどのように進めて
よいのかわからない
具体的に

テスト技法

テスト手法

2

毎イテレーションでの問題をイテレーション完結に情報展開する

イテレーションで見つかったバグをベースにテストでやるところを決め5W1Hにする

システム全体をフォローする活動

各ナレッジ I/Fの整理（仕様ビュー）

バグモニタし．I/F アンマッチを見つける

何をどこまでテストするのかを明確にしておく．

やらないことを明確にしておく．

バグの発生数を予測し．テスト日数やリリース日程を延ばすかを提示する

再テスト時をみじかく最小限で実施

デイリーでデグレ確認（手戻り防止）

バグの傾向調査（あたりどりテスト）

3

仕様(設計のメカニズム)について、
不明なところを
質問することを
何度もやる

QAチームが
感じたリスクを
POやPMに
話す。

設計が学ぶ、
指摘を持っていく
(明らかな設計ミス
など)

勉強する

仕様のもれ抜けを
見つけてくれる
存在
であればうれしい

いっしょに開発している
という風土、感情、教育。

朝会、昼とうで
いっしょに参加して
仲間いしきを
めばえさせる。
日頃からの
コミュニケーション

バックログが
カンバンをながめる
だけでも
いいんじゃないの？