

付録2 アンケート調査票

分類	No	質問	【必須】チームメンバーの人数	【必須】開発スタイル (記入例: 該当する場合、□→■にする)
チーム状況把握	1	アンケート対象のプロジェクトチームについて教えてください		<input type="checkbox"/> アジャイル型 <input type="checkbox"/> ウォーターフォール型 <input type="checkbox"/> その他 ()

7. 強く当てはまる
6. 当てはまる
5. やや当てはまる
4. どちらともいえない
3. やや当てはまらない
2. 当てはまらない
1. まったく当てはまらない

(他に実践しているプラクティスなど)

分類	No.	質問	【必須】回答欄 (数値1～7を記入)	【必須】実践しているアジャイルプラクティス (記入例: 該当する場合、□→■にする)	【任意】コメント欄
予測	1	プロジェクト計画時、顧客やメンバーと話し合い優先順位とリリース時期を明確にした上で計画を作成している 対人的・時間的リソースにリスクは無い? 特定の人しかできない業務はないか? といったことまで考慮している		<input type="checkbox"/> リリース計画ミーティング <input type="checkbox"/> プロダクトバックログ (優先順位付け) <input type="checkbox"/> プロダクトオーナー <input type="checkbox"/> 人材のローテーション	
	2	プロジェクト中にリスクが発覚した場合、具体的な内容をチームメンバー全員が共有・理解した上で作業を進めている 例: 実現性や不明点のリスクなどを事前に把握し、チームで共有している		<input type="checkbox"/> イテレーション計画ミーティング <input type="checkbox"/> イテレーション <input type="checkbox"/> 日次ミーティング	
	3	要件変更があった場合、計画変更を行いチームとして対応が可能な範囲を再定義している 例: 人的・時間的リソースや負荷状況をチームで共有した上で、実現性のある計画を立てている		<input type="checkbox"/> プランニングポーカー <input type="checkbox"/> イテレーション計画ミーティング <input type="checkbox"/> イテレーション <input type="checkbox"/> ベロシティ計測	
監視	4	メンバーの負荷状況やプロジェクトの進捗状況が見える化できており、異常な状態であると判断するメトリクスを設定している		<input type="checkbox"/> かんばん <input type="checkbox"/> タスクボード <input type="checkbox"/> スプリントバックログ	
	5	メンバーが報告しやすい状況・環境が整っている		<input type="checkbox"/> ニコニコカレンダー <input type="checkbox"/> 心理的安全性	
	6	深刻な問題を招く恐れのある状況が見落とされことなく、問題として認識できる		<input type="checkbox"/> 日次ミーティング	
基本能力 対処	7	メンバーが急に減った(例: 入院した)場合、そのメンバーの担当していた作業を短期間で引き継げる		<input type="checkbox"/> ペアプログラミング <input type="checkbox"/> コーディング規約 <input type="checkbox"/> 人材のローテーション	
	8	問題の解決のためにメンバー間の作業調整を行う際、特定のメンバーに作業が集中しないように分担できている		<input type="checkbox"/> 柔軟なプロセス <input type="checkbox"/> 集団によるオーナーシップ	
	9	解決が困難な問題が発生した場合、問題の解決のためにメンバー同士が協力して対応できる		<input type="checkbox"/> 紙・手書きツール <input type="checkbox"/> チーム全体が一つに <input type="checkbox"/> 共通の部屋	
	10	影響の大きな設計変更や機能追加が必要となった場合、柔軟に計画見直しやプロセスの見直しができる		<input type="checkbox"/> プロダクトオーナー <input type="checkbox"/> プロダクトバックログ (優先順位付け)	
学習	11	スキルの高いメンバーが、低いメンバーに対して、日常的に指導している		<input type="checkbox"/> ペアプログラミング <input type="checkbox"/> スプリントレビュー	
	12	開発途中で、今までの作業の振り返りを行い、良い結果をチーム内で共有し、他のメンバーも実践している		<input type="checkbox"/> ふりがえり	
	13	新たな開発手法を取り入れた時、その開発手法の有識者や経験者に直接指導を受けている		<input type="checkbox"/> アジャイルコーチ	
	14	会社やチームがメンバーの学習する環境を積極的に提供している		<input type="checkbox"/> 人材のローテーション	

分類	N o.	質問	【必須】回答欄 (数値1～7を記入)	【必須】実践しているアジャイルプラクティス (記入例：該当する場合、□→■にする)	【任意】コメント欄
技術スキル	15	開発メンバが保有する固有の技術スキルについて、メンバ間での共有(学習・教育含む)が適宜行われている		<input type="checkbox"/> 人材のローテーション <input type="checkbox"/> チーム全体が一つに <input type="checkbox"/> ペアプログラミング	
	16	技術スキルが不足しているメンバでも作業が滞りなく実施できるよう、開発作業に必要な情報はドキュメント化がされている状態である		<input type="checkbox"/> 人材のローテーション <input type="checkbox"/> リアクティング	
	17	将来の開発において必要となりうる技術スキルに関し、チームとして習得に向けた活動(勉強会など)を作業として計画している		<input type="checkbox"/> プロダクトバックログ <input type="checkbox"/> 継続的インテグレーション	
非技術スキル	18	メンバ間での情報共有が活発に行えている		<input type="checkbox"/> 紙・手書きツール <input type="checkbox"/> 共通の部屋 <input type="checkbox"/> かんばん	
	19	問題解決する際、複数の解決案から適切なものを選択できている		<input type="checkbox"/> 柔軟なプロセス	
	20	チーム内で意見が分かれた場合、それぞれの主張をお互い理解してから解決するようにしている		<input type="checkbox"/> チーム全体が一つに <input type="checkbox"/> ファシリテータ (スクラムマスター)	
レジリエンス能力を支える要素	態度	21	特定の人しか知らないことがないように、ソースコードや業務に関する知識を共有している	<input type="checkbox"/> 集団によるオーナーシップ	
		22	チームのメンバは、身勝手な判断や行動ではなく、各自の判断で自律的に動いている	<input type="checkbox"/> 自己組織化チーム	
		23	チームメンバ全員が一つの目標に向かった振る舞いを取っている	<input type="checkbox"/> チーム全体が一つに	
		24	絶えず顧客と相談することで無駄な機能を作らないようにしている	<input type="checkbox"/> オンサイト顧客	
心身の健康	25	メンバの行動や成果物に対して、メンバが相互にフィードバックを適切に実施している	<input type="checkbox"/> 迅速なフィードバック		
	26	メンバは、自分や他のメンバが行っている仕事の目的や重要性を理解している	<input type="checkbox"/> インセプションデッキ <input type="checkbox"/> 集団によるオーナーシップ <input type="checkbox"/> 自己組織化チーム		
	27	メンバは有意義な作業のみを実施し、退屈で繰り返し実施する作業はしなくても良い	<input type="checkbox"/> 自動化された回帰テスト <input type="checkbox"/> 継続的インテグレーション		
	28	無理してあとが続かないやり方でなく、健康的でメリハリのある活動を心がけている。また、メンバーの一人一人が安心して自分らしく働いている	<input type="checkbox"/> ニコニコカレンダー <input type="checkbox"/> 持続可能なベース		

付録3-1 アンケート結果（レジリエンスに対する自己評価）

分類	No.	質問	チーム名																											
			平均	AG_A	AG_B	AG_C	AG_D	AG_E	AG_F	AG_G	AG_H	AG_I	AG_J	AG_K	AG_L	AG_M	WF_A	WF_B	WF_C	WF_D	WF_E	WF_F	WF_G	WF_H	WF_I	WF_J	HY_A	HY_B		
予 測	1	プロジェクト計画時、顧客やメンバーの話し合い優先順位とリソース配分を明確にした上で計画を作成している	5.60	6	6	5	6	6	6	6	5	5	5	5	5	6	5	5	5	5	4	6	6	6	5	7	7	7	7	
	2	プロジェクト中のリスクが顕在化した場合、具体的な内容をチームメンバー全員が共有・理解した上で作業を進めている	5.56	5	5	4	4	7	6	6	7	6	6	4	5	7	5	5	4	5	6	7	5	6	6	4	7	7	6	
	3	緊急変更が発生した場合、計画変更をチームとして対応可能な範囲を再定義している	5.28	6	5	5	4	6	6	4	5	7	6	6	6	7	6	5	5	4	6	7	5	3	5	1	7	5	5	
監 視	4	メンバーの負荷状況やプロジェクトの進捗状況が見えてきており、異常な状態であると判断するメトリクスを設定している	4.96	5	6	5	5	6	6	6	6	3	3	6	6	7	5	6	5	4	3	7	5	3	4	7	7	1	6	
	5	メンバーが報告しやすい状況・環境が整っている	4.96	5	6	5	3	6	6	3	5	4	5	6	6	7	5	6	5	4	3	7	5	3	4	7	7	1	6	
	6	深刻な問題を招く恐れのある状況が見逃されること、問題として認識できる	5.32	6	4	5	5	5	5	5	6	5	2	5	6	7	6	6	4	5	6	7	5	3	4	7	7	6	6	
対 処	7	メンバーが意図した（例：入席した）場合、そのメンバーの担当している作業を短期間で引き継げる	4.60	3	4	4	6	4	2	5	6	2	6	5	4	7	5	7	3	4	5	6	5	4	4	1	6	7	7	
	8	問題の解決のためにメンバー間の作業調整を行う際、特定のメンバーに作業が集中しないように配慮している	4.32	5	5	4	3	4	5	6	6	2	4	3	4	7	3	6	3	5	5	2	5	3	4	1	6	7	7	
	9	解決が困難な問題が発生した場合、問題の解決のためにメンバー同士が協力して対応できる	5.76	5	7	5	6	7	7	6	7	4	6	6	7	7	5	5	4	5	5	7	5	4	5	7	7	5	5	
学 習	10	影響の大きい変更や設計変更や機能追加が必要となった場合、柔軟に計画見直しやプロセスの見直しができる	5.52	4	6	5	5	7	6	5	7	6	6	3	7	7	6	5	5	5	6	4	5	4	6	5	7	7	7	
	11	スキルの高いメンバーが、低いメンバーに対して、定期的に指導している	4.56	2	5	4	6	6	2	4	7	3	4	4	5	6	6	6	2	5	4	7	5	3	5	1	7	5	5	
	12	開発途中で、今までの作業の振り返りを行い、良い結果をチーム内で共有し、他のメンバーも実践している	4.36	5	7	3	1	6	2	3	6	1	7	4	5	7	3	5	4	2	5	5	5	4	5	1	7	6	6	
技 術 的 な 学 習	13	新しい開発手法を取り入れた時、その開発手法の有識者や経験者に直接指導を受けている	3.68	6	4	2	4	4	4	2	5	1	5	1	5	2	3	4	5	3	3	5	4	4	4	1	7	4	4	
	14	会社やチームがメンバーの学習する環境を積極的に提供している	4.84	6	5	3	4	5	3	6	6	7	5	3	6	5	3	3	5	5	3	7	4	5	3	7	7	5	5	
	15	開発メンバーが保有する固有の技術スキルについて、メンバー間で積極的に知識共有（学習・教育含む）が適行されている	4.52	5	5	2	5	6	3	7	6	3	5	4	6	5	6	5	3	5	3	5	4	4	3	1	7	5	5	
非 技 術 的 な 学 習	16	技術スキルが不足しているメンバーでも作業が実施できるよう、開発作業に必要な情報はメンバー間で共有されている	4.52	2	6	5	5	3	3	3	3	5	4	5	3	6	3	4	3	3	4	6	5	5	5	5	7	6	6	
	17	将来的な開発において必要となる技術スキルに関し、チームとして習得に向けた活動（勉強会）を作業として計画している	3.96	6	2	3	2	7	2	7	2	5	2	3	4	7	6	2	3	3	2	3	6	5	4	2	1	7	5	
	18	メンバー間の情報共有が活発に行われている	5.08	4	5	5	5	7	6	6	6	3	5	4	6	7	6	6	4	5	3	7	5	3	2	5	7	5	5	
能 力	19	問題解決する際、複数の解決案から適切なものを選択できている	5.04	3	2	5	4	5	6	5	6	7	7	4	6	7	4	3	3	5	5	6	4	4	4	7	7	7	7	
	20	チーム内で意見が分かれた場合、それぞれの主張をお互いに理解し、協力して解決するおこなっている	5.24	4	5	5	2	6	7	5	5	6	5	6	6	7	6	6	3	5	5	5	4	4	4	7	7	6	6	
	21	特定の人が知らないと知らないように、ソースコードや業務に関する知識を共有している	4.56	3	5	4	5	5	5	4	6	1	5	6	7	7	3	6	3	5	4	6	5	4	4	1	5	5	5	
態 度	22	チームのメンバーは、身勝手な判断や行動ではなく、各自の判断で自律的に動いている	5.00	3	5	5	5	7	6	4	4	7	7	4	7	6	5	5	3	5	4	3	5	5	2	7	6	5	5	
	23	チームメンバー全員が一つの目標に向けた振る舞いを覚えている	5.36	6	5	5	5	7	6	6	6	7	5	6	7	7	5	5	3	5	5	4	5	6	4	1	7	6	6	
	24	絶えず顧客と相談することで無駄な作業を作らせないようにしている	4.56	5	4	5	6	6	7	6	4	1	4	6	5	6	6	5	4	5	4	5	3	4	4	1	3	5	5	
心 身 の 健 康	25	メンバーの行動や成果物に対して、メンバー相互にフィードバックを適切に実施している	4.64	2	4	5	5	5	6	4	6	1	5	4	6	5	6	5	3	5	3	6	4	4	2	7	7	6	6	
	26	メンバーは、自分や他のメンバーが行っている仕事の目的や重要性を理解している	5.44	5	5	6	4	7	6	5	5	6	6	4	6	6	6	6	4	5	5	7	5	5	4	7	7	5	5	
	27	メンバーは有意義な作業のみを実施し、無駄な作業はなるべく減らす	4.24	1	6	4	6	6	6	7	6	3	5	2	6	5	2	3	2	2	3	6	5	4	3	1	7	5	5	
合計	28	無理でも達成したいのではなく、健康でなければならぬことを認識している。また、メンバーのメンバーが自分の役割を認識している	4.92	4	6	6	4	7	4	6	6	1	5	6	6	6	4	5	2	4	4	6	5	4	4	7	4	4		

付録3-2 アンケート結果 (実施しているアジャイル・プラクティス)

予測	合計	AG A	AG B	AG C	AG D	AG E	AG F	AG G	AG H	AG I	AG J	AG K	AG L	AG M	WF A	WF B	WF C	WF D	WF E	WF F	WF G	WF H	WF I	WF J	HY A	HY B
1 リーン計画ミーティング	21	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
2 プロダクトバックログ(優先順位付け)	15	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
3 プロダクトオーナー	6	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
4 人材のローテーション	8	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
5 アイテレーション計画ミーティング	9	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
6 アイレーション	11	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
7 日次ミーティング	21	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
8 ジョニング/ホーカン	2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
9 アイレーション計画ミーティング	9	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
10 アイレーション	12	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
11 ホロシバ計測	6	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
12 かんぽん	9	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
13 タスクボード	10	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
14 スクラムバックログ	11	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
15 ニュートラルメンバー	2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
16 心理的安全	5	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
17 日次ミーティング	20	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
18 バックログマシ	5	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
19 コーディング規範	12	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
20 人材のローテーション	9	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
21 柔軟なプロセス	3	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
22 集団によるオーナーシップ	7	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
23 集約によるオーナーシップ	9	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
24 チーム全体が一括	9	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
25 チーム全体が一つに	11	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
26 共通の部屋	10	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
27 共有の部屋	7	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
28 かんぽん	9	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
29 柔軟なプロセス	8	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
30 チーム全体が一括	9	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
31 ファシリテーター(スクラムマスター)	9	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
32 集団によるオーナーシップ	8	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
33 自己組織化チーム	8	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
34 チーム全体が一括	8	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
35 オンサイト顧客	5	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
36 迅速なフィードバック	2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
37 イベントの生	8	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
38 インベシブティブ	2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
39 集団によるオーナーシップ	7	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
40 自己組織化チーム	8	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
41 自動化された回帰テスト	11	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
42 継続的インテグレーション	10	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
43 ニュートラルメンバー	2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
44 継続可能なペース	8	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
合計	474	16	6	24	31	27	26	31	33	13	25	1	39	40	4	3	10	21	6	4	24	13	1	4	47	25

付録3-3 アンケート結果（コメント）

No	質問	AG A	AG B	AG C	AG D
1	プロジェクト計画時、顧客やメンバーと話し合い優先順位とリリース時期を明確にした上で計画を作成している 対人的・時間的リソースとコストは高いか？特定の人ができない業務はないか？といったことまで考慮している	・開発ロードマップ、開発要件（開発STEP表）、顧客からのヒアリング情報をインプットに、開発計画を決定。 ・緊急度が高く、短期開発案件はアジャイル開発プロセスを適用し、小チームで実施している。 ・短期開発を実現するため、予めスキル開発要件見合うメンバーで構成している。			
2	プロジェクト中にリスクが発覚した場合、具体的な内容をチームメンバー全員が共有・理解した上で作業を進めている 例： 実現性や不明点のリスクなどを事前に把握し、チームで共有している	・メンバー内で日次ミーティングで共有。 （共有はするが、全員の理解を求めている） ・プロジェクト全体のリスクはリーダMTG（週次の上位会議）にて解決している			メンバーおよび顧客でそれぞれ日次ミーティングを実施している。
3	要件変更があった場合、計画変更を行いチームとして対応可能な範囲を再定義している 例： 人的・時間的リソースや負荷状況をチームで共有した上で、実現性のある計画を立てている	要件変更の影響を受けるメンバーを対象に都度計画。 ペロシテ計画が参考になれば活用。			要件変更がある場合は顧客とミーティングを行い、再度優先順位付けを実施して、作業内容を再定義している。
4	メンバーの負荷状況やプロジェクトの進捗状況が見える化できており、異常な状態であると判断するメトリクスを設定している	・1日できる作業量を確認している（逸脱は翌日調整） ・メトリクスなど閾値を設けているわけではない。 ・カンバンツールはTRICHORDを利用		Redmine, JIRA使用。	Redmineによる進捗管理により、メンバー毎の負荷状況を把握している。
5	メンバーが報告しやすい状況・環境が整っている	・若手メンバーを集めて開発チームを構成。 ・互いに気になることを日々コミュニケーションをとっていた。	デイリースクラム、レトロスペクティブで誰でも発言できる雰囲気を感じて作っている		
6	深刻な問題を招く恐れのある状況が見落とされることがなく、問題として認識できる	・日次ミーティングにプロダクトリーダも加わり状況を確認。 ・トピックをプロダクトオーナーへの報告、必要に応じて協議する場がある。			
7	メンバーが急に辞めた（例：入院した）場合、そのメンバーの担当していた作業を短期間で引き継げる	・（1ヵ月で完成させるなど）短期開発のみ適用している。 これまで、メンバーが辞めるなどの状態は発生していない。 ・随時レビューを実施しているので引継ぎなフォローは可能	チーム内で決めている最低限のモデル（クラス図、シーケンス図、状態遷移図）はドキュメントとして残している		メンバー内でのリカバー可能なように標準的なサブローテーションを行うとともに作業標準化のための手順書を作成している。
8	問題の解決のためにメンバー間の作業調整を行う際、特定のメンバーに作業が集中しないよう分担できている	・日次ミーティングで調整。 ・プロダクトとしてはクオリティ/バスのみチェック			
9	解決が困難な問題が発生した場合、問題の解決のためにメンバー同士が協力して対応できる	・日次ミーティング後、随時打合せ ・専用で利用できる開発ルームがあり、必要に応じて集中検討会を開催		Microsoft Teams, Redmine使用。	
10	影響の大きな設計変更や機能追加が必要となった場合、柔軟に計画見直しやプロセスの見直しができる	・開発対象を絞って適用しているため、大きな変更が入れないように進めている。 ・小規模な変更、追加はあるがプロダクト/バックログを活用			
11	スキルの高いメンバーが、低いメンバーに対して、日常的に指導している	・スプリントレビューは実施しているが教育目的には使用していない ・計画段階でプロジェクトチームの技術力を把握し構成している。			
12	開発途中で、今までの作業の振り返りを行い、良い結果をチーム内で共有し、他のメンバーも実践している				
13	新たな開発手法を取り入れた時、その開発手法の有識者や経験者に直接指導を受けている	・初めにアジャイル開発を回す場合は、（社内の）アジャイル開発コーチに参画してもらい、都度助言をもらっている。			最初はM2Mで指導を行っている。
14	会社やチームがメンバーの学習する環境を積極的に提供している	・社内にアジャイル開発入門講座が提供されている。 アジャイルプロセス開発のメンバーは事前に講座を受講させている			関連する技術に対する勉強会などあれば積極的に参加している。
15	開発メンバーが保有する固有の技術スキルについて、メンバー間の共有（学習・教育含む）が適宜行われている	・先端技術、固有技術は（“QU活動”という）チームでの改善活動のフレームワークに載せて、メンバー間で共有を図っている。 ・基礎技術は社内講座を活用。	手順書を作成し展開している		
16	技術スキルが不足しているメンバーでも作業が滞りなく実施できるような、開発作業に必要な情報はドキュメント化がされている状態である	・スキルを補う手段としてのドキュメント化は実施していない ・定型作業は手順書を随時作成している			
17	将来の開発において必要となる技術スキルに関し、チームとして習得に向けた活動（勉強会など）を作業として計画している	No15と同じ。			
18	メンバー間での情報共有が活発に行えている	・非技術スキルについてもNo15の活動を展開している。 ・計画的に実施しているものであり、“活発”に該当するものではない。		Microsoft Teams, Redmine使用。	個人ごとにA4サイズのホワイトボードを利用して、図やシーケンスなどを使用した説明を行っている。
19	問題解決の際、複数の解決案から適切なものを選択できている	■開発段階で発生した問題については ・主導するメンバーの理解に対しメンバー間で協議している（必ず対策を出すという場合はしていない） ■調査検討段階でのまとめの場合は ・必ず複数案を提示し、メリット/デメリットを表にまとめて、結論を出している			
20	チーム内で意見が分かれた場合、それぞれの主張をお互い理解し、そこから解決するようにしている	No.19のやりかた。 ・まとまらない場合は主導メンバーがファシリテータとなって意見を集約し解決案をまとめている			
21	特定の人が知らないことがないように、ソースコードや業務に関する知識を共有している	・レビューは随時実施している。			
22	チームのメンバーは、身勝手な判断や行動ではなく、各自の判断で自律的に動いている	・タスクにして、計画的に作業している。 ・コントロールはできているが自律性は考慮していない			
23	チームメンバー全員が一つの目標に向かって振る舞いを取っている	・短期で開発するものをターゲットとして適用しているため、目標（ゴール）意識は高い			
24	絶えず顧客と相談することで無駄な機能を作らないようにしている	・スプリントレビューに加え、週次で進捗・状況報告を行っている。			日次ミーティングを実施している。
25	メンバーの行動や成果物に対して、メンバーが相互にフィードバックを適切に実施している	・相互にフィードバックするケースは少ない。 ・日次ミーティングで互いの状況は把握している。			成果物を共有するとともに、事前レビューを実施している。
26	メンバーは、自分や他のメンバーが行っている仕事の目的や重要性を理解している	・開発の方向性、目標（ゴール）は理解している。 ・特別なチームのみ適用しているプロセスなのでメンバーの関心や意識も高い。			
27	メンバーは有意義な作業のみを実施し、退屈を繰り返して実施する作業は多くても良い	（質問の意味が受取られない） ・繰り返した作業もタスクには存在する。 ・有意義/退屈と考えるかは本人次第。			自動化されたリグレーション試験を実施している。
28	無理してあとが疲れないやり方ではなく、健康的でメリハリのある活動を心がけている。また、メンバーの一人一人が安心して自分らしく働いている	（質問の意味が受取られない） ・短期集中開発を実現させるためのプロセスとして採用しているの、多少無理は発生する。 ・“安心して”、“自分らしく”と思うかは本人次第。 （プロジェクトの枠組みではメンタルケアは実施していない。労務管理として別枠でケアしている）			

No	AG E	AG F	AG G	AG H	AG I	AG J	AG K
1	少数精鋭で柔軟かつスピーディーに開発を推進しており、多少臨時的になってもリスクが滞りません。				優先順位とリリース時期は明確にするが、顧客ではなく営業は販売会社になる。また、特定の人の業務はないか、などは考えていない。	優先度はつけていて、リリースリスクが無いかは確認しつつ、なかなか厳しい状況が続いている。しかし、どのくらい厳しいのかは、ポイントをつけることでわかっている。	顧客とは話していない
2		Slackおよびかんぱんでの情報共有				毎週の定例会で共有。毎日ではない。	
3							
4	毎日の朝ミーティングでバックログ、かんぱんの状況を踏まえた意思決定を行っており、プロジェクトのヘルシー状態を把握している				作業時間をメンバーに入力して見える化としている。マトリクスや明確な判断基準はない。	チームメンバーがそれぞれ進捗Pを受け持っていることが要因と思われるが、タスクボードを見るだけではなかなか負荷状況が見えてこない。	ニコニコカレンダー
5	コミュニケーションツール (Slack, Mattermost) を導入し、ロケーションや時間に縛られない開発環境を構築。 ※海外出張中も連携可能				定例会で話し合いがない。	(アンケートへのコメント。心理的安全性って、スローカンのように促しているのですが、具体的なアクションになってるでしょうか。)	
6	毎日の朝ミーティングにおいて、各メンバーから課題情報の収集を行っております。	Slackでの情報共有			見落としは少ない。		
7	多少個人化はしているが、共通ライブラリやフレームワーク技術を取り入れるなどして、個々の個性を最小限としている。また、少数精鋭で個々のスキルが高いため、他人の分野でも少し時間をかければメンテナンスすることができる。	コードレビュー			短時間で引き継ぎしていないが、やらない選択はないので、今いるメンバーで何とかする考えになる。	仕事としてバディ制にしているので、急に減った場合にも対応しやすい。また、そのため、プライベートの用事も優先しやすい状況にある。	
8	スピードを優先していることから、得意分野に応じて、バックログをアサインするようにしていることから、多少、特定のメンバーに作業が集中してしまっている。(※若手エンジニアは教育対象のため上級エンジニアに付いて幅広くアサイン)				分散しようとしているが、現実には偏っている。		
9						複数のプロジェクトを並行しているの、同プロジェクト間は協力し合っているが、他プロジェクトまではできない。余裕がない。	共通の部屋でなかったが専用の作業スペースを確保
10						できる日程で調整する。	
11	ベアプログラミングとまでは言えないが、若手エンジニアが上級エンジニアに付いて、レビューを受けながら開発を進めるようにしている	コードレビュー				日常的にはしていない。	ベア設計
12						開発途中の振り返りはやっていない。	
13	文献や外部 (NTTデータ主催) のアジャイルセミナー等に参加し、開発の進め方について勉強している					指導まではしない。	課全員で教育を受ける
14	Udemyの活用 展示会やカンファレンスの参加 ※上記の活動を後押し					提供はしている。	
15	定期的に勉強会を開催。 ※新しい技術の紹介等	コードレビュー	内部勉強会			やろうとするが、現実には忙しくてできない。	互いの成果物やノウハウを定期的に共有する場を設けている
16	少数精鋭で進めることを前提としており、ある程度のスキルレベルが必要となっている					アップデートされていないドキュメントが多い。	
17	先端技術は積極的に試し、勉強会で共有しあっている					計画はされていないし、チームとして取り組もうとしているが現実には忙しくてできていない。	この設問とは関係ないのでは？
18		Slackでの情報共有	Viber, Teams等SNSを用いたコミュニケーション				
19	チーム全体で各家の利点欠点の比較検討を行い、解決策を選択している						KPT (複数のTryを挙げて解決策を検討)
20						ほぼできている	
21		コードレビュー				やろうとするが、現実には忙しくてできない。	
22	少数精鋭で信頼関係を構築して進めている。 開発本来の目的に影響するような部分は、共通の価値観で判断し、チームに相談するようにしている。						
23							
24	こまめにデモを行い、方向性を合わせている					営業や販売会社と直接相談することはほとんどない。	(開発ではないので、設問が対象外かも) 顧客ではなくシステム担当者、ハード担当者とは必ず相談している
25						フィードバックはない。	スプリントレビューのタイミングがメイン。
26	キックオフ資料という形でインセプションデッキを作成し、こまめにアップデートしている					自分の業務の重要性は理解しているが、他のメンバーの業務の重要性は理解できていない。	タイムラインによる長期のふりかえり
27	単体テストおよびリフレクション駆動 (UI結合テスト) を自動化する仕組みを構築。 Jenkins, Github, Selenium					繰り返しの作業も必要ならやる。やらない選択はない。	
28	すでに自動テスト、CI化できたわけではなく、5年間で作り上げたチームである					残業時間が多い。多分不健康。	

No	AG L	AG M	WF A	WF B	WF C	WF D	WF E	WF F	WF G	WF H	WF I	WF J	HY A	HY B
1	WFでは特定の人にしかできない業務があること自体、課題になるという思想がそもそもないので 点数が低い	顧客とは話していない			プロダクトバックログのようなものとして、要件の優先順位をまとめた資料は存在する。			プロジェクト計画レビュー						
2	WFではチームメンバー全員が共有するという思想もないため 点数が低い							朝会や週次の定例会でリスクの共有を行っている						
3	変更対応はWFでもアジャイルでも同じように発生する				要件変更時の計画変更は、顧客と話し合いながら実施できている。				チケット駆動開発			やるしか選択肢はない。		
4	WFでは、メンバーの負荷状況は、見える化することになっていないのでわからない	二ココカレンダー			Redmineを使用している	Redmine, Excelでの日程計画		RedmineまたはExcelによる進捗管理					個人管理 & 非公開	
5	報告しやすいか？ どうかは、リーダ次第のところがあるかも							・メンバー同士が物理的に距離が近い ・コミュニケーションツールの使用				Daily役コミ	1 on 1 ミーティング	
6	朝会をしなくても、リーダが個別に担当者のところに行けば状況はわかるが、WFでは担当者同士で気づく問題の検出ができない													
7	WFの場合、引き継ぎやすい開発をする思想はない				コーディング規約は存在するが、守られていない状態。								代わりはない。	
8	WFの場合、特定のメンバーが集中しない開発をしないという思想はない							チケットベースの作業調整					一部に集中。	
9	WFの場合、協力し合うという思想はそもそもない	共通の部屋でなかったが専用の作業スペースを確保											Teamsでチーム内連携を取っている。	
10	WFは、はじめに決まった要件が前提で最後までやりきるので、開発中の機能変更が発生することは、アジャイルより柔軟にできるやり方ではない				No.3と同様								顧客要求は必達なので、自由度は少ない。	
11	WFは、知識を共有、知らない人に素早く広めたいという意識が、明確になっていない	ペア設計						仕様書レビュー・コードレビュー					暇はない。	
12	WFの振り返りは周期が数か月と長いので、カイゼンサイクルが長い。スクラムの場合、周期が2週間なのでカイゼンサイクルを回しやすい				ふりかえりしても、あまり成果が出ていない								暇はない。	
13	アジャイルの細しさと、良いやり方であることが、チームとして共有できているとは言えない やることになったのでやります みたいなところがある				社内有志者に話を聞いている。								取り入れない。	
14	少しずつはやっているが、ローテーションの方針が明確になっていない				勉強会や展示会を行っている雰囲気。								課長の方針はあるが、暇はないので誰も行動しない。	
15	違いを感じない												暇はない。	
16	現状のアジャイルチームの方はやっていると感じる												一部の人が実施。	
17	現状のアジャイルチームの方はやっていると感じる	この役割とCは関係ないのでは？							勉強会				暇はない。	
18	現状のアジャイルチームの方はやっていると感じる					redmine							日次ミーティングとTeams	
19	違いを感じない												プレストしている。	
20	現状のアジャイルチームの方がやっていると感じる													
21	現状のアジャイルチームの方がやっていると感じる					SVN		ドキュメント化					暇はない。	
22	現状のアジャイルチームの方がやっていると感じる													
23	現状のアジャイルチームの方がやっていると感じる												個々人は別の目標。	
24	現状のアジャイルチームの方がやっていると感じる	顧客ではなくシステム担当者、ハード担当者とは絶えず相談している							事前に販売会社・企画部門と合意した要求仕様に基づく開発				要求は全て必要。	
25														
26														
27													手動のみ。	
28								KI活動					残業規制により、課長以外は早く帰宅している。	

付録4 将来、レジリエンス能力を改善する方法が確立した場合の提案

現状、アンケートなど改善課題が多いが、将来、現場で使えるようになった場合のレジリエンスの改善方法をここに記載する。

方針：自チームの弱点を分析し改善する

<すすめ方>

STEP1：自チームのアンケート調査結果からレジリエンス毎にレーダーチャートを作成する

STEP2：レジリエンス能力の高いチームのデータと自チームのデータを比較する

STEP3：レーダーチャートから、弱点を分析し改善をすすめる。

レジリエンス能力の高いチーム程、レーダーチャートの面積は広がる。

今回は、例としてアンケートの平均点が低いAチームと高いBチームを選択して以下の様に比較した。

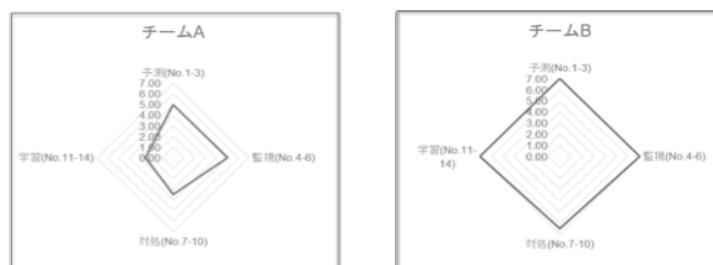


図1 基本能力の比較

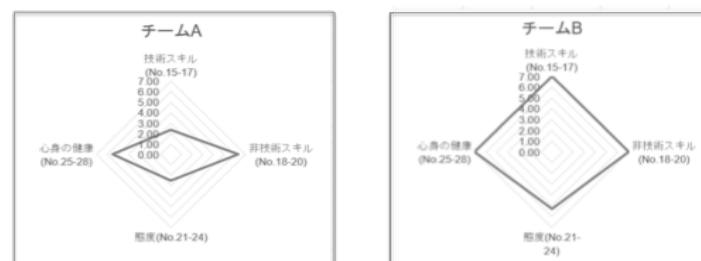


図2 レジリエンスを支える要素の比較

図1より、「学習」に対し Aチーム：2.5, Bチーム：7.0と大きな違い(4.5)があった。

図2より、「技術スキル」に対しAチーム：2.3, Bチーム：7.0と大きな違い(4.5)があった。

点数の低かったAチームでは、いずれも「暇がない」とコメント欄にあったがアンケートの内容（付録2 アンケート調査票No12, 17を抜粋）は、それぞれ、「開発途中で、今までの作業の振り返りを行い、良い結果をチーム内で共有し、他のメンバーも実践している」、「将来の開発において必要になりうる技術スキルに関し、チームとして習得に向けた活動（勉強会など）を作業として計画している」など大きな違いがあった点数のアンケート内容を参考にすることで、レジリエンスの改善へつなげることができるのではないかと考える。

付録5 アンケート高得点チームのプラクティス分析

アンケートの合計点が高得点であった上位2チームが共通して実践しているアジャイル・プラクティスを調べたところ、「プランニングポーカー」、「リファクタリング」、「持続可能なペース」であった。このプラクティスの効果を以下に記載する。レジリエンスが低いチームが今後、どのようなアジャイル・プラクティスを選択するとよいかを考える上でヒントになると考える。

プラクティス	効果
プランニングポーカー	メンバーの業務経験や知識を生かすことができる
	ゲーム感覚で楽しく活発な意見が引き出せる
リファクタリング	わかりやすくつくり、再利用しやすくすることで、将来へのリスクを軽減する活動になる
持続可能なペース	高負荷やストレスが高い状況が続く状況を放置しない
	ゆとりと活気をもって効率的に仕事ができるチームをめざす