

Understanding the InnerSource Checklist

Chapter 9 - Appendix - 日本語訳

Silona Bonewald (著), 茂田井 寛隆 (訳), 小林 良岳 (訳), 加藤 慎介 (訳)

Last Change:26-Jun-2020.

目次

1. 実際のチェックリスト (The Actual Checklist)	2
1.1. 準備 (Readiness)	2
1.1.1. 個人 (Personal)	2
1.1.2. プロジェクト (Project).....	2
1.1.3. チーム (Team)	3
1.1.4. 企業 (Company)	6
1.2. リポジトリ (Repository).....	9
1.3. ソフトウェアツール・システム (Tools).....	10
1.4. プロジェクト実施の計画 (Logistics).....	11
1.5. 製品責任者 (Product owners)	11

この文書は"Understanding the InnerSource Checklist, Chapter 9"を意識したものです。

NOTE

1. 関連URL <http://innersourcecommons.org/checklist/>
2. 原書 <https://innersourcecommons.org/assets/files/InnerSourceChecklist.pdf>

1. 実際のチェックリスト (The Actual Checklist)

InnerSource プロジェクトを開始する事に対してプレッシャーを感じるのはその通りで、ましてや、真の InnerSource またはオープンソース企業への移行を通じて、あなたの組織をかじ取りをするのは簡単ではありません。PayPal社は、独自の経験と移行を行っているほかの企業の同僚の経験からチェックリストを記載しています。私たちは、あなたが何をすべきかを整理し、集中するのに役立つよう、ここにチェックリストを提示します。もちろん、このチェックリストに素直に従うだけでは、成功は得られません。全ての組織は、調査、議論、InnerSource の恩恵を得るための文化的に変化することを始めていく必要があります。しかし、あなたがこのチェックリストを常に確認していれば、進捗があることがわかり安心するでしょう。

(It's easy to feel overwhelmed with the task of starting an InnerSource project, much less taking your organization through the transition to a true InnerSource or open source company. PayPal has drawn a checklist from its own experiences and the experiences of colleagues in other companies making the transition. We present it here to help organize and focus what you need to do. Certainly, following the checklist slavishly will not produce success. Every organization needs to undertake investigations, discussions, and cultural change in order to benefit from InnerSource. But keeping this checklist in front of you can help reassure you that you're making progress.)

この文書は、GNUマニフェストとチェックリストマニフェストの両方に触発されたものです。私たちはあなたがInnerSourceにむけて少なくとも小さな一歩を踏み出し、あなた自身のチェックリストを作成し、それを私たちにも共有(展開)してくれることを期待しています。Apache Wayはチームにとって重要なインスピレーションであり、あなたもこのリストに同じような感覚を感じることでしょう。

(This report was inspired by both the GNU Manifesto and Checklist Manifesto. We hope to inspire you to take at least a small step toward InnerSource and creating your own checklist and sharing it with us. The Apache Way was an important inspiration for the team, and you will spot similarities in this list.)

1.1. 準備 (Readiness)

1.1.1. 個人 (Personal)

- これが会社にとって実行可能な戦略とっていますか？それとも、観念的な誓約と理想論から抜け出そうと活動していますか？
(Do you believe this is a viable strategy for your company, or are you just doing it out of ideological commitment and idealism?)
- InnerSourceで成功するために必要な変化を理解していますか？
(Do you understand the changes required to be successful at InnerSource?)
- あなたは、他人にアイデアを提示したり、相手の心配事に耳を傾けたり、偏りのない共通点を見つけたりするのが得意ですか？
(Are you skillful at presenting ideas to others, listening to their concerns, and finding common ground without bias?)

1.1.2. プロジェクト (Project)

- このプロジェクトは会社にとって重要ですか？戦略変更を受け入れられそうですか？
(Does this project matter to the company? Is it likely to survive strategy changes?)

- 妥当性 (Appropriateness)
 - このプロジェクトは、オリジナルの開発チーム以外の開発者にも興味をもってもらえそうですか？
(Is this project likely to be interesting to developers outside the original development team?)
 - それは今時点で社内の他のチームが依存するほど広く使われている、もしくは広く使われそうですか？
(Is it currently or could it be used widely by other teams in the company who depend on it?)
 - オリジナルチームが予想できなかったような形で、他の開発者による拡張によって利益を受け入れることができますか？
(Will it benefit from being extended by outsiders in ways that the original team could not anticipate?)
 - 他のチームがバグ修正やリファクタリング作業によって貢献することで、プロジェクトはその利益を受け入れることができますか？
(Could the project benefit from having other teams contribute bug fixes and refactoring work?)
 - 他チームの開発者が有用なコードや提案を提供してくれますか？
(Could outside developers contribute useful code or suggestions?)
 - 他チームの開発者が貢献の要請に応えてくれますか？
(Would outside developers respond to appeals to contribute?)
- コードの成熟度 (Code maturity)
 - プロジェクトは簡単にかつ安全に変更が行えるよう十分なモジュール化がされていますか？
(Is the project modular enough to make changes easy and safe to make?)
 - コードは十分にドキュメントがそろっていますか？
(Is the code well documented?)
- 既存の工程 (Existing process)
 - リリースを頻繁に行うことができますか？
(Can releases be made frequently?)
 - 継続的なテストや統合が行われていますか？
(Is continuous testing and integration in place?)
 - 全てのコードがGitHubのようなバージョン管理リポジトリに登録されており、ブランチやプルリクエスト、マージが簡単にできるようになっていますか？
(Is all of the code stored in a version control repository such as GitHub that makes branches, pull requests, and integration easy?)

1.1.3. チーム (Team)

- チームメンバーがInnerSourceの課題に挑戦する準備ができていますか？
(Are team members ready for the challenges of InnerSource?)
 - 他チームの開発者からのコードやコード変更を受け入れられますか？
(Accepting code and changes to their code from outsiders)
 - 他チームの開発者が投稿した完ぺきではないコードに責任を持てますか？

(Being responsible for less-than-perfect code contributed by outsiders)

- あまり完ぺきとは言えないコードを見てもらえる他チームの開発者がいますか？
(Having outsiders see their less-than-perfect code)
- 他チームの開発者の貢献の受け入れや拒否について、それらが時々困難であり会議しなければなりません
(Having to conduct conversations that are sometimes difficult with outsiders about accepting and rejecting their contributions)
- 貢献者への助言し、助言できるようになるよう学べますか？
(Mentoring and/or learning to mentor contributors)
- チームメンバーはInnerSourceプロジェクトを実行するための要件を理解していますか？(メンバーがオープンソースプロジェクトに参加したことがあると望ましい)
(Do team members understand the requirements of running an InnerSource project? (It helps if members have participated on open source projects.))
 - ゲスト貢献者のためのドキュメントの作成・保守
(Creating and maintaining documentation for guest contributors)
 - フォーラムに参加する意思、根気強く質問に回答する意思
(Willingness to participate in forums and answer questions patiently)
 - 廊下での会話の代わりに、誰もが見ることができる履歴付きのフォーラムの使用(オプションで、ミーティング最中にノートをとることができ、全ての決定とその説明が文書化される)
(Using forums, which provide a historical record everyone can see, instead of hallway conversations (optionally, using a scribe to make notes during live meetings, just as long as all of the decisions and explanations for those decisions are documented))
 - 外部からの提出物のコードレビューを行う意思
(Willingness to do code reviews of outside submissions)
 - バグトラッカーのメンテナンス
(Maintaining the bug tracker)
 - Trusted Committer の役目を交代できる
(Taking a turn in the role of TC)
- Trusted Committer を選択したことがありますか？
(Have you chosen TCs?)
 - 彼らは自分たちの責任を理解していますか？
(Do they understand their responsibilities?)
 - GitHubのCONTRIBUTING.mdファイルに名前を書き、保守する
(Write and maintain the CONTRIBUTING.md file in GitHub)
 - 投稿されたコードのレビュー(プルリクエスト対応)
(Review incoming code (pull requests))
 - ゲスト貢献者への助言
(Mentor guest contributors)

- プルリクエストのマージ
(Merge pull requests)
- モジュール化とリファクタリングを率先して実施
(Take the lead on refactoring and modularization)
- 議論リストを使った質問の共有と回答
(Participate and answer questions on discussion lists)
- お知らせ・告知
(Send announcements)
- 連携の機会を見極めと提案
(Watch for and suggest opportunities for collaboration)
- 彼らはこのポジションへの潜在的(内在的・仕事関連)な報酬を理解していますか？
(Do they understand the potential rewards (intrinsic and job-related) of this position?)
 - コードベースへの理解度を深める
(Develop deeper understanding of the codebase)
 - あなたのチームのコード品質を改善する
(Improve the quality of your team's code)
 - より良い統合により、組織内のコード品質が全体的に改善する
(Improve the quality of code within the organization as a whole with better integrations)
 - 多くの投稿されたコードサンプルを見ることにより、開発者として学び成長する
(Learn and grow as a developer by seeing many incoming code examples)
 - 外部からの貢献を促進するためのコードのリファクタリングやモジュール化の方法を理解する
(Understand how to better refactor and modularize code to encourage external contributions)
 - ゲスト貢献者への助言や交渉を通じて、対人能力やリーダーシップ能力を養う
(Develop interpersonal and leadership skills through mentoring and negotiation with guest contributors)
- これまでの職務内容に無いことをするための余裕が与えられていますか？
(Have they been given leeway to do things not previously in their job descriptions?)
- チームメンバーの仕事のスケジュールには、これら新しい責任のための時間が確保されていますか？
(Is time set aside in team members' work schedules for these new responsibilities?)
- チームメンバーはこれら責任を履行するための訓練を受けていますか？
(Are team members trained to handle these responsibilities?)
- 組織内の誰もがフォローしたり、検索したりできるようなアナウンスの仕組みがありますか？
(例： Slack, E-mail)
(Is there a mechanism for making announcements that anyone in the organization can follow and search? (Examples: Slack, email))
- 全てのゲスト貢献者からの質問やチーム内の決定事項が新たなゲスト貢献者から検索できるように、議論を記録しておける仕組みがありますか？(例:Slack, オンラインフォーラム)
(Is there a recorded mechanism for discussion so that all Guest Contributor questions and internal

team decisions are searchable by incoming Guest Contributors? (Examples: Slack, online forum))

1.1.4. 企業 (Company)

経営層 (CxO-level executives)

- 幹部はInnerSourceとしてプロジェクトを運営する目的と価値について理解していますか？
(Do the executives understand the purpose and value of running a project as InnerSource?)
 - 組織全体に*民族的な知識*を広めていく価値
(The value of spreading tribal knowledge across the organization)
 - 独自に抱えている拒絶意識やボトルネックを取り除くチームを持つ価値
(The value of having teams remove their own external blockers and bottlenecks)
 - より相互に繋がった組織の価値
(The value of a more interconnected organization)
 - コードベースの多くの機能をより深く理解している高度なチームメンバーを持つことの価値
(The value of having advanced team members that have a deeper understanding of many arms of the codebase)
 - InnerSourceが、基本的な優れたプラクティスを推奨し、さらに優れたコーディング&開発プラクティスの形成と浸透支援をどの程度行うか
(How InnerSource encourages basic good practice and helps develop and spread even better coding and development practices)
 - InnerSourceが個々の開発者の学習と開発力をどの程度育てるか
(How InnerSource increases the learning and development of individual developers)
- 幹部は柔軟な仕事の要件や部門を超えた貢献に費やされる時間を喜んでサポートしていますか？
(Are they willing to support flexible work requirements and time spent on cross-departmental contributions?)
- 幹部は実験、失敗、再配置を受け入れることができますか？
(Can they accept experimentation, failure, and repositioning?)
- 幹部はマネジメントを必要としないキャリアアップのための道筋を喜んでサポートしていますか？
(Are they willing to support a path for career advancement that does not require management?)
- 経営層は、このイニシアチブを支持していますか？
(Does the executive team support the initiative?)
- 会社の目標とKPIが明確に示され、共有されていますか？
(Are company goals and KPIs clearly stated and shared?)
- 会社のビジョンとミッションには関連性があり、最新のものであり、明確であり、尊重され、守られていますか？
(Are the company's vision and mission relevant, up-to-date, clear, respected, and followed?)

人事部 (Human resources)

- InnerSourceの価値観に基づいた昇進のための補修や基準がありますか？
(Are there rewards and criteria for promotions based on InnerSource values?)

- 他部門のプロジェクトに貢献した人への報酬(制度)
(Rewards for people who contribute to projects in other departments)
- 他部門からの貢献を担当した人への報酬(制度)
(Rewards for developers who handle contributions from other departments (TCs))
- コミュニティの役割を尊重し、管理職への昇進を必要としない昇進の道筋が必要
(Need path for advancement that respects community role and does not require moving into management)

組織体制 (Organizational setup)

- 中心で調整してくれるチーム
(Central coordinating team)
 - チームの準備が整っていますか？
(Is the team set up?)
 - InnerSourceに興味を持っているプロジェクトとコミュニケーションをとる準備ができていますか？
そのプロジェクトとチームの準備ができているかをこのチェックリストの基準に評価できますか？
(Is it ready to communicate with projects interested in going InnerSource? Can it apply the criteria in this checklist to ensure that the project and team are ready?)
 - どのようにプロジェクトをInnerSourceとして登録しますか？
(How do projects register as InnerSource?)
- スタッフメンバーは外部プロジェクトに貢献する時間がありますか？
(Do staff members have time to contribute to outside projects?)
- スタッフメンバーはチームの損益の計測と実証を行うリソースが割り当てられていますか？
(Do staff members have resources to measure and demonstrate gains and losses of teams?)
- スタッフメンバーは自身の専門性やモチベーションに基づいてどのプロジェクトに取り組むかを選ぶことができますか？
(Can staff members choose what projects to work on based on their expertise and motivations?)
- 様々な方向からの優れた貢献を評価できる実力主義の理念がありますか？
(Is there a meritocratic philosophy that can appreciate good contributions from all corners?)

開発者 (Developers)

- "会社関連のある"開発者たちは、InnerSourceプロジェクトへ貢献できることを理解していますか？
(Do developers around the company understand that they can contribute to the InnerSource project?)
- その開発者たちは他のプロジェクトへ貢献することの価値を理解していますか？
(Do they understand the value of contributing to other projects?)
 - 外的妨害要因の除去
(Removing external blockers)
 - 他のツールとの統合を自ら構築すること
(Building integrations with other tools themselves)

- 他のチームがどのようにコードを構築しているか例を通じて学ぶこと
(Seeing how other teams structure code and learn from examples)
- その開発者たちは他のプロジェクトへ貢献するためのプロセスを理解していますか？
(Do they understand the process for making contributions to other projects?)
 - 議論への参加
(Joining the discussion)
 - 貢献するためのルールの閲覧
(Reading the contribution requirements)
 - "FAQ/要望ファイル"の閲覧や記入
(Exploring and/or contributing to the Help Wanted file)
 - 告知への署名
(Signing up for announcements)
- その開発者たちはツールの使い方を知っていますか？
(Do they know how to use the tools?)
 - バージョン管理システム (Version control)
 - プログラミング言語 (Programming language)
 - テスト開発用ツール (Test development)
- その開発者たちは、InnerSourceにおける自分たちのチームの役割をサポートする方法を知っていますか？
(Do they know how to support their team's role in InnerSource?)
 - その開発者たちは、生産的にフォーラムに参加できますか？
(Can they participate productively on forums?)
 - 貢献者からの質問への回答
(Answer contributor questions)
 - その方法を選択した理由の明確な説明
(Describe the reasons for choices that have been made in a way that is clear)
 - フィードバックに対する建設的な対応
(Respond constructively to feedback)
 - その開発者たちは、自分たちのコードのレビューに関する意見交換会に参加できますか？
(Can they participate in dialogs around reviews of their code?)
- その開発者たちは自部署以外のプロジェクトへ貢献することが許可されていますか？
(Are they permitted to contribute to projects outside their departments?)
 - その開発者たちは自分たちのプロダクトオーナーやマネージャからサポートを得る方法を理解していますか？
(Do they understand how to get support from their product owners and managers?)

1.2. リポジトリ (Repository)

- リポジトリに紐づいて、下のような"補助的なリソース"がセットアップされていますか？
(Are ancillary resources set up, such as the following?)
 - ドキュメント (Documentation)
 - 議論用のフォーラム (Discussion forum)
 - バグトラッカー (Bug tracker)
 - Wiki (Wiki)
- 以下のファイルが置かれていますか？
(Are the following files in place?)
 - README.md (まず初めにお読みください.md) (README.md)
 - プロジェクト名 (Project name)
 - このプロジェクトの以前の名前やコードネーム (Any earlier names and codenames for this project)
 - プロジェクトの説明 (Project description)
 - チームリーダーのプロジェクトマネジメント(組織)の連絡先 (Team lead and PM/PMO contact information)
 - 検索目的のキーワード (Keywords for search purposes)
 - 人々がこのプロジェクトを名前で検索できるように (So that people can find this particular project by name)
 - 人々がこのプロジェクトをやっていることで検索できるように (So that people can find this project by what it does)
 - 告知リストへの登録方法 (How to sign up for the announcement list)
 - 議論用のフォーラムのURL (Location of discussion forum)
 - 他関連リポジトリ(例:JIRA, Rally, Confluence)のURL (Location of other repositories (Examples: JIRA, Rally, Confluence))
 - CONTRIBUTING.md (貢献について.md) (CONTRIBUTING.md)
 - 必須 (Required)
 - 目次 (Table of contents)
 - Trusted Committer の名前と連絡先 (Names and contact information for TCs)
 - Trusted Committer の利用可能なスケジュール (TC availability schedule)
 - 任意 (Optional)
 - コミュニティガイドライン (Community guidelines)
 - コーディング規約 (Code conventions)

- テスト規約 (Testing conventions)
- ブランチ作成規約 (Branching conventions)
- コミットメッセージ規約 (Commit-message conventions)
- 良いプルリクエストを作成するためのステップ (Steps for creating good pull requests)
- 機能リクエストの提出方法 (How to submit feature requests)
- バグレポートの提出方法 (How to submit bug reports)
- セキュリティレポートの提出方法 (How to submit security issue reports)
- ドキュメントの書き方 (How to write documentation)
- 依存関係情報 (Dependencies)
- (自動)ビルドプロセスのスケジュール (Build process schedule)
- スプリントスケジュール (Sprint schedule)
- ロードマップ (Road map)
- 有益なリンク, 情報, ドキュメント (Helpful links, information, and documentation)
- リポジトリのクローズタイミング (When the repositories will be closed to contributions)
- HELPWANTED.md (FAQ/要望ファイル.md) (HELPWANTED.md)
 - 最初は空っぽでも可 (Can be initially empty)
 - 要望や提案が投稿されたフォーラムとリンクを張ることが可能 (Can link to a forum where requests and offers are posted)
 - 要望や提案のリストを含むことが可能 (Can contain a list of requests and offers)
- GETTINGSTARTED.md "(入門書)" (GETTINGSTARTED.md)
 - 貢献者がアプリを立ち上げてコーディングを開始するまでに必要なステップ全て (Whatever a contributor might need to get the app up and running to begin coding)
 - インターンシップ生のような初心者や初めて貢献された方たちが、始めるにあたって自分にとって役に立ったであろうことについてのフィードバックに基づいて、記入が可能 (Can be filled in by an intern, or after some contributors get started, based on feedback about what would have been helpful to them to get started)
- 誰がリポジトリのドキュメントをレビューしますか？
(Who reviews the documents in the repository?)

1.3. ソフトウェアツール・システム (Tools)

- バージョン管理システム (Version control)
- 継続的統合システム (Continuous integration)
- 試験ツール、試験システム (Testing)

1.4. プロジェクト実施の計画 (Logistics)

- スプリント (Sprints)
- コードアソン[コーディングに特化したハッカソン] 特にツール関連 (Codeathons, especially for tools)
- 教育 (Education)
- 試験 (Testing)
- ゲーミフィケーション[参加者の意欲をかき立てる手法] (Gamification)
- レポート作成 (Reporting)
 - 直近のプルリクエストの (Pull requests during recent period)
 - 総数 (Number)
 - ライン数 (Size (in lines of code))
 - 種類 (Type)
 - プルリクエスト送信者の名前 (Name of pull request submitter)
 - ベースラインの指標 (Baseline metrics)
 - 継続的なトラッキング (Ongoing tracking)
 - 人的・物的リソース (Resources)
 - 費用 (Costs)
 - 時間 (Time)
 - 測定結果を誰が閲覧するか？コミュニティへ開示するか？ (Who sees measurements? Display to the community?)
- バグ修正 (Bug fixing)

1.5. 製品責任者 (Product owners)

以降、記載なし。

以上