Sycomor methodological diagnosis : a case study

1. Introduction

This document offers a synthetic overview of a typical application of our methodological analysis to a practical case. While this is not intended to be a full-fledged analysis and our mission report will provide additional detail, all steps of a complete audit are nevertheless explicited and summarized, as is a realistic course of action to be undertaken in order to improve development productivity, product quality and team efficiency.

2. Industrial context

X[1] is a start-up company specializing in medical diagnosis based on a proprietary technology for biomarker identification. It employs ca. 80 people. Its activity is split between two major departments: Service (technology is applied to customer-provided biological samples) and R & D (research for new biomarker signatures and development of diagnosis kits). Although its core business is strongly grounded in biotechnology, the company had the foresight to create and maintain a bioinformatics department for the explicit purpose of providing solutions to internal needs: management of produced biodata; result visualization and management tools made available to their customers; applied data analysis research for the identification of new signatures.

The Service Department processes are bound to undergo an ISO 9001 labelling in the middle term; at least part of the bioinformatics developments will be significantly impacted and need to submit to the same quality criteria. In order to prepare for these coming changes and in a more general move toward higher quality and productivity made almost mandatory by the ever increasing volume of processed data, the Chief Information Officer (CIO) decided to hire Sycomor for a methodological diagnosis.

3. Mission overview

<u>Audit</u>: based on the company's size and the number of interested parties, the audit is scheduled to last about 5 days; the Chief Scientific Officer (CSO), Operations Lead Manager (representing the Service Department), and CIO are first asked to share their high-level vision of their needs and goals during a group meeting (½ day), then a more in-depth individual interview (1 day total). Afterwards, the Information Systems and Bioinformatics team participate in a simillar collective audit (½ day), before shorter individual interviews are conducted (3 days), the latter allowing us to perform a global skillset review for the development teams.

This audit step has the obvious benefit of giving Sycomor a better grasp of the various activities performed by the Bioinformatics department. As can be expected, based on the complexity and heterogeneity of business processes, the team is involved on several levels:

---

1. For the sake of confidentiality, the company we describe here is fictitious. However, all descriptive elements are based on real studies conducted by Sycomor.

i. Low-level management of data production and systems interoperability (transfer and handling of raw biodata);
ii. Dedicated internal information management system for managing experiment traceability (identification, description, progress, validation, reporting), storage and quality control;
iii. User-friendly, interactive result visualization application provided to customers for acces to and management of their data;
iv. Prototyping and automation of innovative analysis methods.

For historical reasons (increasing involvement of the department with very different parts of the business activity without a long-term unifying strategy), active technologies are very heterogeneous and implementation is mostly project-dependent: (i) Perl, Python, awk scripts; (ii) Java web-application ; (iii) Java standalone application. (iv) R scripts and embedded Java code.

Analysis : following these interviews, and additional specific Q&A sessions with relevant parties, a synthesis can be achieved, which will summarize the identified issues, highlight possible ways of improving the development process and provide a suggested course of action in order to benefit from them. The next sections of this document give a brief overview of this summary, based on the methodological process carried out by Sycomor.
Fore more info, please refer to http://www.sycomor.fr/en/methodo.html.

4. Structural diagnosis

Organizational Axis: Interviews allowed us to discover that R&D bioinformatics analysts are “embedded” in the biotechnological research projects they are associated with and lack both a clear link to the Bioinformatics department hierarchy and regular communication channels with their peers. Since their job still entails some *ad hoc* coding, this results in a significant number of redundant developments among them.
Course of action :

- Creation of a “meta-project” team including all analysts, scheduling of regular meetings to help disseminate information, share knowledge and pool development resources.
- Creation of a “Bioinformatics Liaison” profile with both technical and scientific background and skills; she will directly report to the Development team but will participate in analyst meetings, offer her support on the technical side of their activity (notably when analyst-developed prototypes need to be migrated to a more industrial process) and provide feedback to the Development team on all analysis-related topics.

Human Resources Axis : our skill review allowed Sycomor to ascertain that team members all display solid skills in their respective fields but, due to their training and diverse background have an uneven grasp of good programming practices (GPP) and development methodologies. Furthermore, their technical skillsets are very heterogeneous, which leads to a strong coupling between the developer and its project, resulting in a distinct lack of mobility: this is a major risk whenever the need for knowledge transfer arises (new priorities, turnover...). A largely untapped advantage of the hybrid profiles displayed by the team is the presence of a wide array of connex skillsets (statistics, biology, algorithmics, computer science...) and expressed topics of interest, some members embracing the informatics aspect of their job, while others are inclined to nurture a more versatile approach.

Course of action :

- Group training sessions aimed at enforcing a minimal basic knowledge of all priority technologies across the team; this will help inter-project mobility and encourage knowledge and skill transfers.
- Creation of a “Tech Referee” profile, with a strong background in software engineering, who will have the added responsibility of creating, monitoring and encouraging technology watch initiatives as well as providing internal seminars and training sessions on emergent technologies and promising tools.
- Explicit definition of a “personal projects time”, alowing developers to focus on more personal side projects as long as they are deemed potentially useful to the company (but not necessarily a priority) : technology watch, exploratory developments, technical experiments, analysis projects, data science... This is a good way to empower employees, allow them to explore and maintain their connex skills, and stay innovative and creative.

<u>Project Management Axis</u>: the fragmented nature of the team, structured around micro-groups each dedicated to a single specific development project, has not encouraged the emergence of sufficiently mature project management methodologies. Until now, this structural weakness was less sensitive, with groups working independently, but the need for more robust processes, the growing complexity of user change requirements and the increasingly demanding delivery cycles are about to make it a critical bottleneck. Basics (notably code versioning repositories) are thankfully already present, but only partially used (no strict delivery process; no versioning of analysis prototypes).
Course of action :

- Implementation of a simple, integrated, user-friendly management solution (Redmine) for the centralizaion of versioning repositories, project monitoring, user change requirements, bug reporting...
- Formulation of a more frequent and synthetic reporting across all teams.
- Extension of versioning procedures to all developments, including protoypes.

<u>Technological Axis</u>: technological heterogeneity is a major roadblock toward better sharing and pooling of knowledge and expertise. Moreover technology watch, although present, remains limited and has never been marked as a priority, resulting in significant technological backwardness that simple caution toward emergent technologies cannot justify alone.
Course of action:

- Controlled migration of all developments to the Java stack, since it is already integrated in some projects and is able to satisfy all identified development needs.
- Internal training with and evolution toward recent technological upgrades (JEE6, JSF2, Java7, JPA 2, Spring).
- Increased technology watch centering on third-party libraries (e.g. Guava) and computational performance improvements (parallelizaion, Hadoop, Spark...).

5. **Operational diagnosis**

<u>Methodological Axis</u>: the aforementioned historical reasons (fragmented development groups, technological heterogeneity, lack of training) have lead to insufficient application of development methodologies. We were able to identify that development cycles (variable duration and scope,

random deliveries) were especially lacking in a robust and accepted structure. This also extends to mediocre quality control, resulting in excessively long and stressful post-delivery and “patching” sequences.
Course of action:
- Definition of short (1 to 3 months) development cycles with a limited, explicit scope and subject to user validation.
- Improvements on estimation of development requirements, especially time management (finer-grained development blocks, increased monitoring).
- Explicit formulation of user interactions and communication channels (contractual validation of development cycle goals to ensure greater stability and less change requirements; integration of users in intra-cycle change definitions and beta testing).
- Definition of explicit progress and success criteria, including some quantitative indices and qualitative feedback from users (satisfaction surveys).
- Generalized code review specifically targeted at redundancy reduction (refactoring, development pooling, general usage libraries) and collective development skill improvements.
- Definition of adequate test coverage (partial for the general code base, but complete for critical processes and components backed by third-party libraries susceptible to change).
- Definition of good delivery practices (monitoring, improved usage of version repositories, rational version numbering, delivery notes, packaging and deployment automation, addition of a pre-production server with daily deployment and quality control).

<u>Programming Axis</u>: the implementation of good programming practices (GPP) was left up to each developer, resulting in a fragmented code base of variiying quality, in terms of readability, robustness and performance.
Course of action:
- Training and internal knowledge sharing for a better dissemination of GPP (naming conventions, documentation...).
- Installation and systematic application of static analysis tools (PDM, CheckStyle...), with the definition of a custom company-wide ruleset and quality metrics.
- Incremental review and correction of the existing codebase to make it comply with these new quality constraints.
- Creation of regularly scheduled code reviews, hotseating and pairing, especially for test definition and coding.

<u>Modelling Axis</u>: a quick evaluation of modelling practices in use by the development team revealed that existing data models are robust, but generally built on the relational application side, thus limiting access to some fundamental object-oriented features (inheritance). Moreover, modelling is usually performed too quickly and focuses on static views, resulting in multiple revisions along the development lifecycles to overcome unanticipated pitfalls and limitations.
Course of action:
- General adoption of a forward engineering approach, making modelling the root of all developments (followed by semi-automated code and databse generation).
- Increased efforts on modelling and UML practices so as to include at least some dynamic use cases.

Analytical Axis: the data analysis team boasts strong skills in statistical learning, supervised classification and statistics. This last evaluation axis is therefore limited to some simple suggestions aimed at improving knowledge management and its benefits to end users.

Course of action:

- Selection of ontologies allowing for a formal description of business processes and entities of interest.
- Integration of these semantic resources to the platform, as a source of structured annotations and research keywords.
- General use recommandations for result annotation and visualization (keyword tagclouds, navigation across annotation links and annotation similarity measures).