Journal of Management Science and Business Intelligence, 2025, 10–1
July. 2025, pages 01-03
doi: 10.5281/zenodo.15869970
http://www.ibii-us.org/Journals/JMSBI/
ISBN 2472-9264 (Online), 2472-9256 (Print)

An analysis of project provenance through a novel method of framework analysis

Larson, T.[1, *] and Crouse, G.

[1]Department of Computer Science, Oklahoma State University

*Email: bjorn.larson@okstate.edu

Received on 5/21/2025; revised on 7/10/2025; published on 7/12/2025

Abstract
Subsequent to the US removal of non-compete agreement protection for companies has a primary impact in the realm of intellectual property. This paper proposes that one substitute for the circumstantial evidence of hiring an employee currently under the terms of a non-compete agreement with a competing organization is through the use of digital forensics. One novel approach for this may be in the form of analysis by way of categorizing milestones and other data points and comparing the relative stages of any given project planning methodology. The methodology proposed is the SDLC, as a wide-ranging and foundational model.

Keywords: SDLC, Intellectual property, project management, analysis

1 Introduction

Intellectual property is directly protected under US Patent law (US Code, Title 35, 2023). The specifics of what constitutes intellectual property, theft of intellectual property, and the specific protections therein are presumptive for this paper and left to the course of law. However, presuming that there has been a challenge between two parties (the accuser, Company A, and the accused, Company B), one accusing the other of patent infringement based on the trade secrets presumptively covered by legal protections, substantiating the accusation is a difficult matter.

The historical workaround was the establishment of so-called "Non-Compete Agreements" wherein Company A would require, as a condition of employment, that their employees would not work for any competing companies, with various provisions and caveats, for a set period of time. The assumption was that, by the time that the period covered by the NCA was over, any specific and protected knowledge that the employee had would be out-of-date, protected through other legal means, or otherwise moot.

The question arises as to why these protections were necessary in the first place. Given the scenario in which both Company A and Company B are developing products in secret, it is feasible that Company B would respond to any accusations by Company A with the defense 'we were working on the same thing at the same time, and because of the secret nature of the projects, we were completely unaware of their progress'. If we assume that Company B was, in fact, engaging in a form of corporate espionage, there would be little way for Company A to prove it. Or, alternately, the proof would be expensive and time-consuming to the point that both Company A and Company B recognize the utility of the contrivance of industry-standard NCAs.

However, the FTC (2024) has recently ruled, effectively, that NCAs are unenforceable as a matter of law. The result has been an increase (TSLB, 2024) in the frequency of "trade secret litigation", based on the Defend Trade Secrets Act (2016). The result is frequently expensive and lengthy (AIPLA as cited in Fisher Phillips, 2023).

To this extent, we developed a framework (Crouse & Larson, 2025) whereby organizations can reduce the individualized nature of any given forensic analysis and standardize the process and the findings.

2 SDLC

The Software Development Lifecycle, also frequently known as the Systems Development Lifecycle is "A formal or informal methodology for designing, creating, and maintaining software" (NIST, 2022). In short, it is a guideline, adaptable, and prone to fit whatever standard is needed. Without going into the history of the progenitors to the SDLC, it can be safely said that the SDLC is the standard project management methodology in the industry, with the understanding that it is an umbrella method under which other monolithic and agile methods derive (Saravanos & Curinga, 2024, Alazzawi et al., 2023).

Broadly, the distinction between phenotypes of the SDLC is not so much in the process or the application, so much as the distinction between phases. Here are three classic examples:

1. Waterfall Style – This is the version that directly adapts the nomenclature of one of the precursor versions, which idiomatically suggested that like going over a waterfall (or cascades), it's easier to start over the sooner you stop.

a. Analysis
b. Design
c. Implementation
d. Testing
e. Maintenance

2. Classic Style – This is a revised version specifically returning to the idea of risk-exposure balancing from the waterfall method, and that testing as a distinct phase should be discretized prior to implementation (this is not saying that an implemented solution isn't tested, just like the prior version didn't imply a prototyping activity).

a. Analysis
b. Design
c. Prototyping
d. Implementation
e. Maintenance

3. Modern, Post-Formalized Methodologies style – This is a modern 'kitchen sink' approach that attempts to aggregate all of the extant phases into one overarching cycle.

a. Planning
b. Analysis
c. Design
d. Prototyping
e. Implementation
f. Testing
g. Deployment
h. Design

The selection of the Classic style was used for three primary reasons:

1. The authors' familiarity with it as well as its more simplistic overview.

2. It doesn't matter which one is used, so long as the same comparisons are made between them. Again, none of the presentations of the same SDLC are changing the specific tasks (or their order) that are being done, just how they're aggregated into phases. The proposed method should also be relatively self-normalizing.

3. There is some concern that the prevalence of formalized, custom methodologies is causing a certain amount of partisanship between structured methodologies that is avoided by sticking with the more structured method.ot.

3 Postulate 1 - Deliverables

For the purposes of this paper, each phase will also be given a named deliverable. In essence, this is the conceptual 'thing' that is being passed as input to the subsequent phase.

Analysis – This phase represented ideation, determination of scope, estimate of budget allowance, overview of the current situation, and identification of areas in need of improvement. In new projects, those areas in need of improvement are likely 'this project doesn't exist, yet!' whereas in continuing projects, it should be clear from issues that arose during the previous cycle. The deliverable from this phase is the Problem Statement.

Design – this phase represents brainstorming, structured problem solving, organization, role assignment, research, experimentation, solution proposals, and any other industry-specific steps that may need to be done in order to explain how to get to a proposed solution. The deliverable from this phase is the Design Document.

Prototyping – This can be considered a 'testing' phase and has historically had a number of alternate solutions. But, the substantive majority of the time a small-scale version of the final solution is tried and evaluated, so prototyping is largely an industry-standard practice. The deliverable from this phase is the binary Go-No Go binary decision.

Implementation – Implementation is a wide range of sections. Based on the design document and the success of the prototyping phase, it should be the large scale deployment of a solution, the distribution of a product, the building of a tool or location, etc. This phase is largely project-specific. The deliverable from this phase is the Summary Report.

Maintenance – This last phase is largely considered the largest, with estimates based on application being from 50-90% of the time and cost of a project. This leads directly into the next cycle of the SDLC's analysis phase. The deliverable from this phase is the Trouble Ticket.

The deliverable identifiers are convenient handles: not always exactly what they describe so much as whatever it is that a given project happens to use to transition from one phase to the next phase. Regardless of whether the SDLC was used in the planning and execution of the project, a post-mortem analysis should still see each one of those sections. And absence of them would indicate a non-holistic pattern of the project.

The absence of any of these deliverables alone would not be enough to imply intellectual property theft, however we would estimate that a larger framework with more extensive, non-confidential or sealed, data would suggest the following three metrics would be needed:

An accusation prior to discovery or analysis

The absence of artifacts for some or all of the deliverables when paired with the presence by another company.

Similarities or exact copies between the two companies in, at least, one place.

4 Postulate 2 – Timing

For the purposes of this paper, there is a presumption that, should confidential, proprietary, or sealed data be measurable, the data would adhere to standard statistical distributions. For the purposes of speculation, we presume that the data would adhere to a normal curve, with the following properties:

The data is comparable within a given industry or project type, with the data being used to analyze what would constitute an in-industry comparison group.

The data is comparable within project scope, with the scale of the budget being used as the indirect measure.

Given comparable data, we would expect that the time and complexity as measured by the number of milestones or documentation would be clustered around a mean, with a tight standard deviation.

The literature seems to support these precepts, with numerous papers dedicated to predicting the time a given project would take based on the above mentioned factors (Putnam & Myers, 1992, Bayram, 2016, Guo e al., 2019), with one interesting source suggesting that theoretical models have little practical evidence behind them (Varajão et al., 2022). In short, this is not a new idea, however this phrasing of it is still presumptive.

Secondarily, these models, as suggested by Varajão are all predictive models. It is likely that the exact timing of a wide range of projects across industries will never be made available due to several factors: the sealed nature of post-lawsuit/settlement disclosure, the exact timing and methods being labelled as trade secrets, and the lack of capacity for long-term longitudinal studies in information systems research where speed and agile methodology are commonplace.

Thus, Varajão turns out to be the most accurate source in suggesting that models need to have data to back them up, which will lend themselves towards bespoke methodologies produced by investigative bodies.

5 Hypothesis – Comparability of Adherence

Thus, if we accept the two postulates as being presumptively true, we can say that: for a given project in a given field with a given budget range, its time and cost and tradeoffs between time and cost will be statistically comparable to other projects in the same field so long as the same methodology (i.e. SDLC) is used between them such that the deliverables between each phase are unimportant aside from the fact that these deliverables exist and are documented. So, we hypothesize that we can use these properties to develop a framework that will identify fraud or IP theft within a project by comparing timestamps of deliverables and relative costs between the various deliverables.

This is, again, supported by several studies. Verenich (2018), for example, suggests that it is difficult to predict the timing of the completion of a project based on the "unique attributes and contextual factors" of each project. However, we are suggesting that for sufficiently large projects and/or a sufficiently large data set, and organization could overcome these through data analysis.

6 Conclusion

IP theft has always been an issue. Companies have a variety of reasons to rely on trade secrets rather than patent protection in order to maintain their competitive advantage in certain aspects. However, it is within the intention and framing of much of the nature of intellectual property law to allow for trade secrets, and trade secrets are explicitly a protected form of intellectual property.

With the removal of the protections of Non-compete Agreements (side note: we are not making a moral or legal position on the nature of NCAs, simply the fact that they were in use and now are unenforceable), it becomes increasingly important for organizations who want to maintain in fact, and not just in right, their protection under the law to utilize empirical and standard methods of identifying violations.

References

Alazzawi, A., Yas, Q., & Rahmatullah, B. (2023). A Comprehensive Review of Software Development Life Cycle methodologies: Pros, Cons, and Future Directions. Iraqi Journal for Computer Science and Mathematics. 4. 173-190. 10.52866/ijcsm.2023.04.04.014.

Bayram, Savas. (2016). Duration prediction models for construction projects: In terms of cost or physical characteristics?. KSCE Journal of Civil Engineering. 21. 1-12. 10.1007/s12205-016-0691-2.

Crouse, G. & Larson, T. (2025). Determining the provenance of projects based on phase analysis of software development (U.S. Patent Application 63/773.737). U.S. Patent and Trademark Office.

Defend Trade Secrets Act of 2016, Pub. L. No. 114-153, 130 Stat. 376 (2016). https://www.congress.gov/114/plaws/publ153/PLAW-114publ153.pdf

Federal Trade Commission. (2024). Non-compete clause rule. Federal Register, 89(87), 30001-30035. https://www.federalregister.gov/documents/2024/05/07/2024-09171/non-compete-clause-rule

Fisher Phillips. (2023, May 25). 3 major points to consider when deciding whether to sue over restrictive covenants and trade secret violations. https://www.fisherphillips.com/en/news-insights/3-major-points-to-consider-when-deciding-whether-to-sue-over-restrictive-covenants-and-trade-secret-violations.html

Guo, J., Hu, C., & Bao, R. (2019). Predicting the Duration of a General Contracting Industrial Project based on the Residual Modified Model. KSCE Journal of Civil Engineering. 23. 10.1007/s12205-019-1543-7.

National Institute of Standards and Technology. (2022). Secure Software Development Framework (SSDF) Version 1.1. https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-218.pdf

Putnam, L. H., & Myers, W. (1992). Measures for Excellence: Reliable Software On Time, Within Budget. Yourdon Press.

Saravanos, A. & Curinga, M. (2023). Simulating the Software Development Lifecycle: The Waterfall Model. Applied Systems Innovation, 6(6).

Trade Secrets Law Blog. (2024, September 13). The rise of trade secret litigation. Seyfarth Shaw LLP. https://www.tradesecretslawblog.com/2024/09/the-rise-of-trade-secret-litigation/

United States Code, Title 35—Patents. (2023). https://www.govinfo.gov/content/pkg/USCODE-2023-title35/html/USCODE-2023-title35.htm

Varajão, J., Lourenço, J., & Gomes, J. (2022). Models and methods for information systems project success evaluation - A review and directions for research. Heliyon, 8(12), e11977. https://doi.org/10.1016/j.heliyon.2022.e11977

Verenich, I., Dumas, M., La Rosa, M., Maggi, F., & Teinemaa, I. (2018). Survey and cross-benchmark comparison of remaining time prediction methods in business process monitoring. arXiv preprint arXiv:1805.02896.