

## Lista dos 15 artigos explorados

1. Automated framework for classification and selection of software design patterns - HUSSAIN, Shahid et al. Automated framework for classification and selection of software design patterns. *Applied Soft Computing*, v. 75, p. 1-20, 2019.
2. Design Pattern Elicitation Framework for Proof of Integrity in Blockchain Applications  
- SAEEDI, Kawther et al. Design Pattern Elicitation Framework for Proof of Integrity in Blockchain Applications. *Sustainability*, v. 12, n. 20, p. 8404, 2020.
3. Feature-Based Software Design Pattern Detection  
- NAZAR, Najam; ALETI, Aldeida. Feature-Based Software Design Pattern Detection. arXiv preprint arXiv:2012.01708, 2020.
4. Software design pattern mining using classification-based techniques  
- Dwivedi, Ashish Kumar, Anand Tirkey, and Santanu Kumar Rath. "Software design pattern mining using classification-based techniques." *Frontiers of Computer Science* 12.5 (2018): 908-922.
5. A survey report of enhancements to the visitor software design pattern  
- Pati, Tanumoy, and James H. Hill. "A survey report of enhancements to the visitor software design pattern." *Software: Practice and Experience* 44.6 (2014): 699-733.
6. Software design pattern mining using classification-based techniques  
- Dwivedi, Ashish Kumar, Anand Tirkey, and Santanu Kumar Rath. "Software design pattern mining using classification-based techniques." *Frontiers of Computer Science* 12.5 (2018): 908-922.
7. A survey of experienced user perceptions about software design patterns  
- Zhang, Cheng, and David Budgen. "A survey of experienced user perceptions about software design patterns." *Information and Software Technology* 55.5 (2013): 822-835.
8. Design patterns selection: An automatic two-phase method  
- Hasheminejad, Seyed Mohammad Hossein, and Saeed Jalili. "Design patterns selection: An automatic two-phase method." *Journal of Systems and Software* 85.2 (2012): 408-424.
9. Enhancing the Student Learning Experience in Software Engineering Project Courses  
- Marques, Maira, et al. "Enhancing the student learning experience in software engineering project courses." *IEEE Transactions on Education* 61.1 (2017): 63-73.
10. An intelligent tutor for teaching software design patterns  
- Berdun, Luis, Analía Amandi, and Marcelo Campo. "An intelligent tutor for teaching software design patterns." *Computer Applications in Engineering Education* 22.4 (2014): 583-592.
11. Applying Software Design Patterns in Electromagnetic Field Simulators [EM Programmer's Notebook]  
- Zhu, Guangran Kevin. "Applying Software Design Patterns in Electromagnetic Field Simulators [EM Programmer's Notebook]." *IEEE Antennas and Propagation Magazine* 54.2 (2012): 174-179.
12. A tool for design pattern detection and software architecture reconstruction  
- Fontana, Francesca Arcelli, and Marco Zanon. "A tool for design pattern detection and software architecture reconstruction." *Information sciences* 181.7 (2011): 1306-1324.
13. Identification and Assessment of Software Design Pattern Violations  
- Abdelaziz, Tamer, et al. "Identification and Assessment of Software Design Pattern Violations." *النشرة* 13-6 : (2019) 1.2 *المعلوماتية في الحاسبات والمعلومات*.
14. A cloud service implementation for evaluating design pattern in software evolution  
- Ting, Der-Hong, et al. "A Cloud Service Implementation for Evaluating Design Pattern in Software Evolution." *J. Inf. Sci. Eng.* 31.3 (2015): 1051-1070.
15. Persuasive software design patterns for social influence  
- Oduor, Michael, Tuomas Alahäivälä, and Harri Oinas-Kukkonen. "Persuasive software design patterns for social influence." *Personal and ubiquitous computing* 18.7 (2014): 1689-1704.

**16. Forming and Assessing Student Teams in Software Engineering Courses**

-Løvold, Henrik Hillestad, Yngve Lindsjörn, and Viktoria Stray. "Forming and assessing student teams in Software Engineering courses." *International Conference on Agile Software Development*. Springer, Cham, 2020.

**17. Understanding coordination in global software engineering: A mixed-methods study on the use of meetings and Slack**

-Stray, Viktoria, and Nils Brede Moe. "Understanding coordination in global software engineering: A mixed-methods study on the use of meetings and Slack." *Journal of Systems and Software* 170 (2020): 110717.

# Automated framework for classification and selection of software design patterns

HUSSAIN, Shahid et al. Automated framework for classification and selection of software design patterns. **Applied Soft Computing**, v. 75, p. 1-20, 2019.

## 1. Fichamento de Conteúdo

O artigo inicia-se contextualizando que abordagens de Linguagem de Modelagem Unificada (UML), Ontologia e categorização de texto têm sido usadas para automatizar a classificação e seleção de padrões de projeto. O artigo propõe uma estrutura para superar a limitação das técnicas de automação existentes, utilizando de técnicas de aprendizagem não supervisionadas que são usadas para explorar o framework proposto. O objetivo do framework proposto é a qualificação e a seleção dos padrões de projeto, sendo assim foi proposto um modelo de avaliação para avaliar a eficácia da estrutura proposta. Foi avaliado a eficácia da estrutura em três aspectos: 1 - o classificação de padrões de projeto, 2 - identificação de classe de padrão de projeto apropriada, 3 - a seleção de padrões de projeto adequados para um problema de projeto. A fim de avaliar a eficácia da estrutura para a classificação, foram usados catálogos de padrões de projeto de domínios diferentes. Tal como, no caso do design da Gang of Four (GoF) catálogo de padrões, a eficácia da estrutura proposta pode ser determinada através da classificação de 23 padrões de design no número apropriado de classes de padrão.

## 2. Fichamento de Bibliográfico

- Gang of Four (GoF) conjunto de soluções e princípios que ajudam os desenvolvedores a criar software (Página 8)
- Design pattern collections é um conjunto de padrões de projetos. (Página 8)
- Douglass pattern collection aplicações em tempo real, o padrão de design Douglass é publicado e categorizado em cinco classes, nomeadamente Memória, Recursos, segurança e confiabilidade, simultaneidade e distribuição. (Página 8)

## 3. Fichamento de Citações

- “The term design pattern class is coined for a group of design patterns which are suggested for a design problem in the proposed framework. Subsequently, the effectiveness of the proposed framework is assessed in terms of unsupervised learner’s decisions regarding the recommendation of a number of pattern classes.”
- “Like supervised learners evaluation, precision and recall measure can be used to assess the effectiveness of unsupervised learners. However, we need to assume that members (i.e. design patterns) of the same class are more relevant as compared to rest of members, and relevancy can be determined using similarity measures.”
- “He fats development of software design patterns. Consequently, patterns organization is required to facilitate the software developers. This situation increases the importance for patterns organization with respect to text classification.”
- “Preprocessing is the first phase of the proposed framework and is applied to problem description of all design patterns and the design problem. ”

# Design Pattern Elicitation Framework for Proof of Integrity in Blockchain Applications

SAEEDI, Kawther et al. Design Pattern Elicitation Framework for Proof of Integrity in Blockchain Applications. *Sustainability*, v. 12, n. 20, p. 8404, 2020.

## 1. Fichamento de Conteúdo

O artigo inicia-se contextualizando como a tecnologia blockchain tem o potencial para transformar práticas convencionais de forma eficiente e dinâmica. Porém ele cita a dificuldade para migrar para o blockchain devido à complexidade de sua infraestrutura e processos. Então esse artigo busca estudar os padrões de projetos para aplicativos blockchain, buscando reduzir a complexidade na compreensão e construção de aplicativos em blockchain. Ele fornece uma demonstração do padrão de Proof of Integrity (PoI) obtido a partir de duas aplicações diferentes em blockchain. A aplicabilidade do padrão é avaliada pela construção de um aplicativo blockchain para verificar a integridade dos certificados acadêmicos e explicando como esta integridade tem sido alcançada na prática. O artigo conclui que blockchain é uma tecnologia em evolução que garante uma plataforma segura para aplicativos distribuídos. Porém, essa nova tecnologia é complexa. Sendo assim, identificar uma empresa caso que pode ser implementada com sucesso em blockchain e também identificando o recurso de padrões do aplicativo blockchain é difícil.

## 2. Fichamento de Bibliográfico

- Blockchain é uma razão distribuída que contém um conjunto de blocos de dados sequenciados. Cada bloco registra os dados da transação de forma transparente, imutável e segura. (Página 2)
- Proof of Integrity (PoI) é comumente observado nos aplicativos que fornecem aos usuários a capacidade de verificar se um determinado documento, registro, acordo ou contrato é autêntico. (Página 2)
- Aplicativos distribuídos são aplicações projetadas para executar em mais de um computador. (Página 14)

## 3. Fichamento de Citações

- “The participation layer is responsible for setting the layout features of the blockchain. It includes different types of aspects that influence the design decision and scope of participation with the blockchain. These layers include consensus, permissions, and control.”
- “The power of blockchain lies in its secure and distributed architecture. It offers a secure decentralized platform for peer-to-peer transactions without the need for a trusted central authority. Blockchain complements the current technologies to integrate conventional business practices with a new level of efficiency and security.”
- “The challenges of adopting blockchain applications are also highlighted as these propel the present research to the goal of defining the design pattern for blockchain applications.”
- “The computing refers to the computing resources that host the blockchain, such as a high security business network, cloud, and data center. The block is the place where the transactions data are stored. The data are stored in the blocks in hashed form”

# Feature-Based Software Design Pattern Detection

NAZAR, Najam; ALETI, Aldeida. Feature-Based Software Design Pattern Detection. **arXiv preprint arXiv:2012.01708**, 2020.

## 1. Fichamento de Conteúdo

O artigo começa relatando sobre como os padrões de projetos de software são soluções comuns em projeto e arquitetura de software. Ele relata que a detecção manual de padrões de projeto é uma tarefa demorada e desafiadora. Por isso, pesquisadores propuseram técnicas de detecção automática de padrões de projeto para facilitar os desenvolvedores de software. Porém essas técnicas apresentam um desempenho baixo para determinados padrões de projeto e esse artigo apresenta uma abordagem que visa melhorar o desempenho. Para isso, são usados recursos de código com classificadores de aprendizado de máquina para treinar automaticamente uma detecção de padrão de design. Foi criada uma representação semântica do código-fonte dos recursos de código e do gráfico de chamadas, e aplicado o Algoritmo Word2Vec. Essa abordagem modela e identifica padrões de projeto de software com 78% de precisão e 76% de recuperação. Foi realizada uma comparação e os resultados mostraram que essa abordagem supera as abordagens de benchmark em 30% e 10%.

## 2. Fichamento Bibliográfico

- Algoritmo Word2Vec é um método para aprender eficientemente um Word Embedding independente, a partir de um corpus de texto. (Página 2)
- Machine learning é uma evolução do estudo de reconhecimento de padrões e da teoria do aprendizado computacional. (Página 2)
- Arquitetura de software consiste na definição dos componentes de software, suas propriedades externas, e seus relacionamentos com outros softwares. (Página 6)

## 3. Fichamento de Citações

- “Design pattern detection is a subjective process and in most cases, the decision of selecting a design pattern is based on the context it is used.”
- “Recently, neural network based models have received increasing attention for their ability to compute dense, lowdimensional representations of word. To compute such representation, i.e., the word embeddings, several models rely on a huge amounts of natural language texts from which a vector representation for each word is learned by a neural network.”
- “Thus, supervised classification learning corresponding to the ground truth that is a class contains a design pattern or does not contain a design pattern can be used to predict design patterns.”
- “ Existing datasets labelled with design patterns are either too small or not publicly available. Therefore, we create a new corpus DPD, which we label with the respective design pattern.”

## Forming and Assessing Student Teams in Software Engineering Courses

LØVOLD, Henrik Hillestad; LINDSJØRN, Yngve; STRAY, Viktoria. Forming and assessing student teams in Software Engineering courses. In: **International Conference on Agile Software Development**. Springer, Cham, 2020. p. 298-306.

## 1. Fichamento de Conteúdo

O artigo inicia-se contextualizando que em projetos de desenvolvimento de softwares um fator importante é o trabalho em equipe. Por isso os cursos de Engenharia de Software estimulam que seus alunos trabalhem em equipes para assim aprenderem como se comportar e as práticas do trabalho em equipe. Sendo assim os instrutores dos cursos encontram vários desafios ao ministrarem para cursos que exigem trabalho em equipe, pois eles devem saber como formar equipe de alunos de alto desempenho. Então esse artigo tem como objetivo avaliar se existe uma diferença no desempenho quando as equipes são escolhidas pelos próprios alunos e quando as mesmas são escolhidas pelo instrutor. Esse estudo foi realizado a partir das equipes participantes de um curso de engenharia de software envolvendo um grande projeto ágil. Esse curso tinha 200 alunos que trabalhavam em 39 equipes, 76,3% dos alunos formavam sua própria equipe, e o restante optou por equipes formadas pelos instrutores. Os alunos foram designados para escrever um aplicativo para o sistema operacional Android envolvendo dados API coletados do Instituto de Meteorologia. O resultado obtido pelo artigo foi de que os alunos que escolheram sua própria equipe acabaram se saindo melhor, devido ao fato de já se conhecerem e já terem estudado juntos.

## 2. Fichamento Bibliográfico

- Algorithm-based tools to automatically match students são algoritmos criados para separar os alunos em equipes e tentar obter um bom desempenho (Página 1)
- API são padrões de programação para acesso a um aplicativo de software.(Página 2)
- themselves são as equipes formadas pelos próprios alunos. (Página 3)

## 3. Fichamento de Citações

- “Early on in the process of designing the course, the question about how teams were to be formed, and how involved in the forming of teams the instructors were to be, arose.”
- “The report and product accounted for 50% of the final grade given to students, the other half being the result of a final individual written exam. The questions on the exam were both from theory presented in lectures and group sessions and from the project they were a part of in the teamwork.”
- “During a project period of 13 weeks, students were assigned to write an app for the Android operating system involving API data gathered from the Institute of Meteorology”
- “In our course, all the teams followed an agile project model. While Scrum was the process model most focused on in the lectures, this was not the most used process model among the teams. Scrum was chosen by 17 teams. However, the majority of the teams incorporated Kanban elements into their Scrum process models.”

**Understanding coordination in global software engineering: A mixed-methods study**

## on the use of meetings and Slack

STRAY, Viktoria; MOE, Nils Brede. Understanding coordination in global software engineering: A mixed-methods study on the use of meetings and Slack. **Journal of Systems and Software**, v. 170, p. 110717, 2020.

### 1. Fichamento de Conteúdo

O artigo trata de uma abordagem para entender melhor os mecanismos de coordenação, estudando sobre reuniões e a ferramenta de colaboração Slack. O artigo conduziu um estudo de caso utilizando os métodos mistos de pesquisa, observação, entrevistas e registro de conversas. Foi estudado o uso do Slack em 4 das equipes de cerca de 30 colaboradores em um centro de produtos da Geosoft, o motivo da escolha por esse centro de produto, era porque eles já estavam maduros em relação a métodos ágeis e já usavam o Slack desde 2015. Na hora das entrevistas, os quatro times foram chamados de front-end, back-end, operações e experiência do usuário. Foi coletado registros do Slack que incluíam aproximadamente 30.000 mensagens enviadas entre membros da equipe. Já na parte de reunião os resultados quantitativos mostraram que os funcionários em projetos globais gastam em média 7h45min por semana em reuniões programadas e 8h54min em reuniões não programadas. O artigo obteve como descoberta que a falta de uma pessoa-chave, a ausência de um suporte organizacional para reuniões não programadas e a atividade desequilibrada dos membros da equipe em reuniões e no Slack, era a principal barreira para uma coordenação eficaz. O ponto positivo encontrado no uso de ferramentas em equipes distribuídas foi o aumento da comunicação informal e diminuição de e-mail.

### 2. Fichamento Bibliográfico

- Slack é uma plataforma proprietária de comunicação comercial desenvolvida (Página 2)
- Mixed-methods são métodos mistos para metodologia que será utilizado para se conseguir obter os resultados (Página 4)
- Meetings são os encontros para tratar de assuntos relacionados ao projeto. (Página 5)

### 3. Fichamento de Citações

- “Meetings in this section, we first present the background and context of the case studied, such as the agile method used and the size of the teams. The content of heavy GSE is important to understand the reported phenomenon.”
- “All transcripts of interviews, observation notes, documents and Blacklogs were imported into NVivo, encoded and discussed among the authors.”
- “Geosoft has projects that are either allocated in one site or shared between two or more locations. One team member or manager typically participates in more than one project.”

## Enhancing the Student Learning Experience in Software Engineering Project

## Courses

Marques, Maira, et al. "Enhancing the student learning experience in software engineering project courses." *IEEE Transactions on Education* 61.1 (2017): 63-73.

### 1. Fichamento de Conteúdo

O artigo trata do método Reflexive Weekly Monitoring que foi criado com o intuito de monitorar equipes em um curso de graduação de engenharia de software, o curso tem foco voltado para realizar na prática os conceitos que são aprendidos pela engenharia de software. Esse curso usa uma abordagem instrucional para a aprendizagem baseada em projetos, isso envolve os alunos em tarefas autênticas do mundo real. As equipes trabalham no projeto por doze semanas, e no final devem implementar a solução. O RWM tem na liderança monitores que realizam sessões de monitoramentos semanais, facilitando assim a compreensão dos membros sobre o status do projeto e o seu desempenho, e destacado no artigo também que esse monitor não é um gerente de projeto, coach ou Scrum master. Após as avaliações concluíram que ambas equipes tiveram limitações de coordenação no primeiro incremento, porém após o segundo incremento essas limitações só foram encontradas em equipes que não tinham monitoramento, e com isso conseguiram suportar a hipótese de que o método RWM impacta positivamente a coordenação entre os membros da equipe.

### 2. Fichamento Bibliográfico

- Reflexive Weekly Monitoring é um método que foi criado com o intuito de monitorar equipes de desenvolvimento, em um curso de engenharia de software (Página 3)
- Scrum master atua como facilitador do Daily Scrum e torna-se responsável por remover quaisquer obstáculos que sejam levantados pela equipe durante essas reuniões (Página 3)
- hipótese é uma formulação provisória, com intenções de ser posteriormente demonstrada ou verificada. (Página 7)

### 3. Fichamento de Citações

- “The RWM method is led by monitors who perform weekly monitoring sessions with a software development team to facilitate members’ understanding of their own performance and project status. The monitor is not a coach, project manager, or Scrum master”
- “The team reflection is designed to make students reflect on possible causes of the issues identified in the team and project diagnosis conducted in the previous step.”
- “To compare the experimental group (monitored teams) to the control group (non-monitored teams), the study used the information gathered using the regular course tracking instruments - the peer assessment conducted at the end of each increment and a statistical report provided by the requirements tracking tool (SRM) [54] used by the students in their projects.”