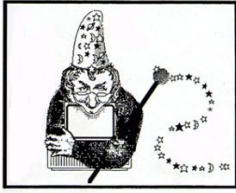


Wicked Problems, Righteous Solutions



A Catalogue of Modern
Software Engineering
Paradigms

Peter DeGrace
Leslie Hulet Stahl

YORKIN PRESS COMPUTING SERIES

Wicked Problems, Righteous Solutions

A catalogue of modern software engineering paradigms

Autores: Peter DeGrace y Leslie Hulet Stahl



El libro incluye un ejemplo detallado de cómo se aplicaría Scrum en el desarrollo de software.

¿Qué aporta?

Trata sobre los problemas complejos y difíciles de resolver en el campo de la ingeniería de software y cómo abordarlos.

Es un libro que aporta claridad en cuanto al origen y la evolución del término “scrum”.

Ideas clave

Los **wicked problems** pueden ser ambiguos, inciertos y estar interconectados. Para resolverlos se requieren **righteous solutions**, paradigmas y enfoques adaptativos y flexibles que se aparten de los métodos tradicionales.

Es en uno de estos enfoques, el **All-at-Once Model** y sus tres versiones, en el que nos encontramos con el concepto “scrum”, anterior a la propuesta de Sutherland y Schwaber.

Temas

Scrum

Comprar libro

En inglés: [Amazon](#)

En español: no disponible

Formación

[Scrum Master](#)

Libros relacionados



Scrum y su origen

El término «scrum» proviene del rugby y se refiere a la formación de los jugadores de rugby al disputar la pelota. No será hasta 1986 cuando Hiroataka Takeuchi e Ikujiro Nonaka lo empleen metafóricamente en la expresión «moving the scrum downfield».

Pese a que se suele situar su origen en 1995 (Jeff Sutherland y Ken Schwaber), el verdadero origen se encuentra cinco años antes, en el capítulo «The All-at-Once Model» de este libro.

The All-at-Once Model

Es un modelo que propone que las actividades de diferentes fases se realicen simultáneamente.

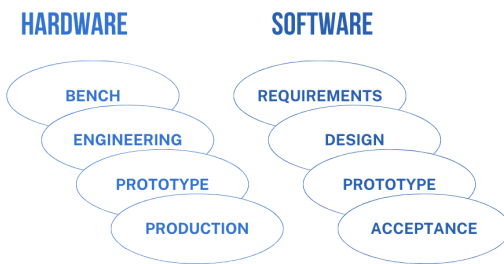
Tiene tres versiones:

- **Scrum/Sashimi:** para equipos.
- **Handcuff:** para dos personas.
- **Hacking:** para una sola persona.

Sashimi

Es un enfoque inspirado en las experiencias japonesas con el modelo *Waterfall*. En este enfoque se reducen las fases tradicionales a cuatro, sin eliminar ninguna actividad, lo que resulta en un solapamiento entre las fases. Esto promueve una comunicación e integración entre las fases y la cooperación entre las fases contiguas.

Se diferencia del modelo *Waterfall* porque permite una interacción más fluida entre las fases, en lugar de seguir un flujo lineal y secuencial.

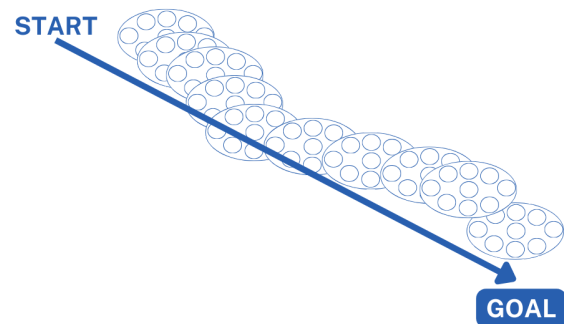


Sashimi Approach

Scrum

Es un enfoque evolutivo del modelo Sashimi. Reduce el número de fases, de cuatro a una y todos los miembros del equipo colaboran estrechamente para avanzar en el proyecto, similar a cómo los jugadores en una formación de scrum en rugby trabajan juntos para mover el balón por el campo.

En lugar de dividir el proyecto en fases separadas con diferentes equipos trabajando en cada una, Scrum propone formar un equipo multidisciplinario con representantes de cada área de experiencia, que colaboran juntos a lo largo de todo el proyecto. Esta colaboración continua y cercana en todas las fases del proyecto ayuda a abordar problemas complejos de manera más eficiente y adaptativa.



Scrum Approach

Nuevos paradigmas de ingeniería

Los **problemas «wicked»** se caracterizan por ser difíciles de definir y resolver, debido a su complejidad y su naturaleza cambiante. Son únicos, y resolver uno puede crear otros.

Para resolver estos problemas el enfoque tradicional no es adecuado, por lo que se exploran distintos modelos y paradigmas que puedan considerarse **«righteous solutions»**. Deben ser soluciones adaptativas y capaces de manejar la ambigüedad y el cambio, basadas en la iteración y la retroalimentación.

Whirlpools

Es una versión compleja del modelo *Waterfall*. Propone que las actividades en el desarrollo de software están interrelacionadas de manera compleja, y estos remolinos de actividad pueden afectar el progreso del proyecto de manera significativa.

Las cuatro fases principales del remolino son: Análisis, Diseño, Codificación y Pruebas. Estas fases están en un flujo constante y pueden suceder simultáneamente o iterativamente, afectando y siendo afectadas por las otras fases.

Spiral Model

Es un modelo incremental que tiene dos versiones: una desarrollada por Boehm y otra por Boeing. Combina la idea de desarrollo iterativo con los aspectos sistemáticos y controlados del modelo de cascada. Se organiza en una serie de ciclos o "espirales", cuyo centro son los recursos necesarios para llevar a cabo el proyecto.

Conforme el proyecto avanza y las espirales van formándose, el crecimiento se representa por los cambios en el diámetro de las espirales.

Prototyping

Es un método más informal del desarrollo de software. Comienza con un protosistema y unos requisitos incompletos. El desarrollador cambia el proto-sistema para alcanzar los requisitos, los verifica y, si aparecen más, repite el proceso. En caso contrario, completa la documentación y termina el prototipo.

El prototipado se utiliza fuera del modelo *Waterfall*: o bien antes para limitar los riesgos; o después para extender el desarrollo y el mantenimiento. Pero también puede utilizarse dentro del modelo *Waterfall* para conseguir mejores requisitos, o incluso puede sustituirlo total o parcialmente.

Prototipar tiene sus beneficios: nos da más control sobre el problema y permite la cooperación de los usuarios. Sin embargo, no tiene buena reputación ni documentación y a veces puede producir sistemas con un bajo rendimiento.

Otros modelos

Hay cuatro modelos que también contribuyen a solucionar problemas complejos:

- **Video Models:** aquí se incluyen *The Hollywood Model*, *The Outside-in Model* y *The MTV Model*.
- *The Cleanroom Model*.
- *User Computing*.
- *System Engineering*.

