HelixSpecifier

HelixSpecifier

التطوير الموجَّه بالمواصفات الذي يضبط طقوسه بنفسه بما يتناسب مع حجم العمل.

الملخص

HelixSpecifier هو محرك Go — سير عمل التطوير الموجَّه يدمج ثلاث منهجيات تطوير
بالمواصفات من SpecKit، وانضباط التطوير الموجَّه بالاختبارات من Superpowers، ودورة حياة
GSD — الإنجاز المعتمدة على المعالم من في تدفق تكيفي واحد. يصنّف المحرك كل مهمة حسب
مستوى الجهد ويتكيف مع حجم العمليات وفقاً لذلك.

وصف مختصر

HelixSpecifier هو محرك دمج للتطوير الموجَّه بالمواصفات مخصص لوكلاء AI. يجمع بين
SpecKit وSuperpowers وGSD، يشغل مراحل يصنّف العمل حسب مستوى الجهد،
المواصفات المدعومة بالمناقشات، يفرض نسبة دنيا بين الاختبارات والكود، ويتعلم من كل تدفق مكتمل.

وصف مفصل

HelixSpecifier هو محرك دمج للتطوير الموجَّه بالمواصفات (SDD) مكتوب بلغة Go (الوحدة
`digital.vasic.helixspecifier`)، ومصمم ليكون مكوناً من مجموعة HelixAgent لوكلاء
AI. يأخذ ثلاث ممارسات تطوير تعيش عادةً في أدوات منفصلة — وعقول منفصلة — ويدمجها في
SpecKit سير عمل تكيفي واحد: عملية التطوير الموجَّه بالمواصفات المكونة من سبع مراحل في
(Constitution، التنفيذ، التخطيط، المهام، التحليل، توضيحها، تحديد المواصفات)، وانضباط التطوير
مع التنفيذ المتوازي للوكلاء الفرعيين، وإدارة دورة حياة Superpowers الموجَّه بالاختبارات في
كل ركيزة تستمر في أداء ما تجيده؛ أما المحرك فهو ما يجعلها GSD. الإنجاز المعتمدة على المعالم في
تعمل كتدفق متماسك بدلاً من خط أنابيب مخيط يدوياً.

يصنّف المحرك العمل الوارد حسب مستوى الجهد ويتكيف مع :الطقوس التكيفية الفكرة المركزية هنا هي
حجم العمليات بما يتناسب معه، بحيث لا تُجرّ تصحيحات بسيطة من سطر واحد عبر الطقوس الثقيلة
نفسها التي تمر بها الميزة الرئيسية — ولا تمر الميزة الرئيسية بمرونة تصحيح الأخطاء المطبعية. عشر
Constitution"ميزات رئيسية تُبنى على هذا الأساس: التنفيذ المتوازي للمهام مع حدود متزامنة، و
نايكيست" الذي يتتبع ويفرض نسبة دنيا TDD"ككود" قابل للقراءة آلياً يفرض القواعد الإلزامية تلقائياً، و
بين الاختبارات والتنفيذ، ومناقشات متعددة الجولات بين عدة وكلاء لتحسين المواصفات، وتعلم التكيف

مع مستويات المهارة، وتحليل الكود القديم، واستخلاص المواصفات التنبؤية من الأنماط التاريخية، ونقل المعرفة عبر المشاريع، وضبط الطقوس أثناء التشغيل لإعادة ضبطها أثناء التنفيذ، وذاكرة مواصفات دائمة مع بحث دلالي.

يُستخدم المحرك كوحدة Go — عبر `go get` خلف محرك — صغير API أو توجيه استبدال محلي الحجم عن قصد: تسجيل الركائز الثلاث بالإضافة إلى مقياس الطقوس وذاكرة المواصفات، وتصنيف جهد العمل، ثم تنفيذ التدفق الكامل والحصول على نتيجة مصنفة بالجودة. السطح بسيط؛ أما التنسيق وراءه يُطوّر المحرك تحت نظام تحقق مضاد للخداع، مع Helix، فليس كذلك. وكما هو الحال مع بقية عائلة مشغل تحديات داخلي يمارس الكود الحقيقي بدلاً من النماذج الوهمية.

لماذا بنيناه

التطوير الموجَّه بالمواصفات، والتطوير الموجَّه بالاختبارات الصارم، وإدارة الإنجاز المعتمدة على المعالم هي عادةً ثلاث ممارسات منفصلة تستخدم ثلاث أدوات منفصلة. بُني HelixSpecifier حتى من تشغيل الثلاث كتدفق متماسك ومتكيف ذاتياً بدلاً من خياطتها يتمكن وكيل AI (HelixAgent) يدوياً.

المحتوى

لماذا يُعدّ ثورةً في المجال

إنه يجعل العملية متناسبة مع حجم العمل — تلقائياً. عادةً ما تعلق الفرق عند أحد طرفي نقيض سيئين: إما طقوس صارمة لكل شيء (آمنة لكنها بطيئة، ومكروهة في صمت)، أو عدم وجود طقوس على هذا التناقض من خلال ضبط حجم الطقوس HelixSpecifier الإطلاق (سريعة حتى تتعثر). يزيل وفقاً لجهد كل مهمة مصنّف، ويعيد ضبطها أثناء التنفيذ مع كشف العمل عن نفسه. القدرة التي لم تكن ممكنة من قبل هي عملية تُضبط تلقائياً لكل مهمة — وفوق ذلك، قرارات المواصفات مدعومة بمناظرة متعددة الجولات والوكلاء، مع تقييم المواقف بدلاً من الاعتماد على تخمين أولي لوكيل واحد.

ما الجديد

- **طقوس تكيفية** — مستوى العملية مدفوع بمقاييس الجودة في الوقت الفعلي ويُعدل أثناء التنفيذ، وليس ثابتاً مسبقاً.
- **اختبار التنمية وفق مبدأ نايكويست** — بوابة نسبة الاختبار إلى التنفيذ (الحد الأدنى 2x)، مستوحاة من منطق نظرية أخذ العينات لنايكويست: لالتقاط السلوك بدقة، يجب أخذ العينات بمعدل أعلى بكثير من معدله، لذا يجب أن تتجاوز الاختبارات الكود الذي تغطيه.
- **بنية المناظرة** — تحسين المواصفات عبر جولات متعددة ووكلاء متعددين، حيث تُقترح المواقف وتُقيّم وتُجمع، لتحل محل الرأي الواحد بمناظرة تنافسية.
- **النقل عبر المشاريع** و**المواصفات التنبؤية** — يستخرج المحرك الأنماط المتراكمة لتوقع المواصفات ونقل المعرفة المكتسبة بشق الأنفس من مشروع إلى آخر.
- **ككود Constitution** — قواعد المشروع الإلزامية قابلة للقراءة آلياً وتُفرض بواسطة المحرك، بدلاً من تركها لليقظة البشرية.

أكبر التحديات التقنية وكيف تغلبنا عليها

- أن GSD وSuperpowers وSpecKit تفترض — **دمج ثلاث منهجيات دون صراع بينها** كل منها يمتلك سير العمل. تم حل المشكلة بمحرك دمج يسجل كل ركيزة خلف واجهة مشتركة ويدفعها عبر دورة حياة تدفق مشتركة، بحيث تتكامل في عملية واحدة بدلاً من ثلاث عمليات متصادمة.
- إذا بالغت في التقدير، يتباطأ كل شيء؛ وإذا — **تحديد مقدار العملية التي تحتاجها مهمة معينة** قللت، تُشحن الأعمال الخطرة دون فحص. تم حل المشكلة بمصنف للجهد يضبط حجم العمل يغذي مقياساً للطقوس يعدل مستوى العملية ديناميكياً أثناء التنفيذ.
- تم حل المشكلة باستبدال — **الحفاظ على جودة المواصفات دون حاجز بشري لكل قرار** المواصفات أحادية الجولات بمناظرة مدعومة بالتكرير، حيث يُقيّم الوكلاء المواقف المتنافسة عبر الجولات، وبفرض نسب اختبار التنمية وفق مبدأ نايكويست حتى لا يتفوق التنفيذ على اختباراته.

حزمة التقنية

- **Go** — اختيرت ليُصدر المحرك كملف ثنائي قابل للاستيراد دون تبعيات وقت التشغيل؛ نموذج التوازي فيها هو ما يجعل إرسال المهام بالتوازي المحدود وجولات المناظرة متعددة الوكلاء ممكنة دون تعقيدات التزامن.
- **logrus** — تسجيل منظم مُنسّق عبر المحرك والركائز الثلاث، بحيث تصبح قرارات التدفق مقروءة بعد التنفيذ (التصنيف، تغييرات الطقوس، نتائج المناظرة).
- **ركيزة SpecKit** — عملية تطوير موجهة بالمواصفات ذات المراحل السبع (Constitution تحديد المواصفات → توضيح → تخطيط → المهام → تحليل → تنفيذ)، لتشكل العمود الفقري المنضبط لكيفية تحول المواصفات إلى كود.
- **ركيزة Superpowers** — انضباط اختبار التنمية مع تنفيذ الوكلاء الفرعيين بالتوازي، لتوفر الصرامة في الاختبار أولاً والتفرع الذي يحافظ على صدق التنفيذ وسرعة أدائه.
- **ركيزة GSD** — إدارة المراحل والمعالم، لمنح التدفق إحساسه بـ"الانتهاء" وتقدمه عبر المراحل.
- **متجر ذاكرة المواصفات** — فهرس دائم وقابل للبحث الدلالي للمواصفات السابقة، وهو الأساس الذي يجعل المواصفات التنبؤية والنقل عبر المشاريع ممكناً بدلاً من البدء من الصفر في كل مرة.

المحتوى

ملاحظات الحالة والنزاهة

- **الحالة: تجريبية.** HelixAgent مُستهلَك كوحدة مكوِّنة من Go ضمن `specifier`.
- **الرخصة: لم تُحدَّد بعد.** لم يُكشف عن أي رخصة عبر GitHub API — غير مُتحقق منها / غير مُصرَّح بها.
- الاسم المعروض "HelixSpecifier" يُحيل إلى المستودع `specifier`.

درجة الأولوية: Helix-أساسية.